



# Accelerating GSK's Open-Source Journey with Gen AI

Andy Nicholls and Christina Fillmore

Technology

## ChatGPT sets record for fastest-growing user base - analyst note

The  
Economist

Menu

Weekly edition

The world in brief

Search

Graphic detail | Daily chart

## ChatGPT could replace telemarketers, teachers and traders

Here's why that is no bad thing

# Introduction

## LLM's in Biostatistics from today to tomorrow

- How GSK has adapted to the new technology today
- What we have experimented with
- Where we are thinking of going in the future
- Share the experience we gained along the way



## Starting with Today



# The Power of LLM's

- The number of ways to directly use LLM's are nearly endless
  - Emails
  - Code generation
  - Learning new programming languages



# The Start of Our Journey with LLM's

- Internalised GPT model, available for all to use
  - Thanks to our partners in GSK's AI/ML team

AN what is the {stringr} function that will trim white space from a character string? ✎



The function in stringr package that will trim white space from a character string is ``str_trim()``.

Prompt: Querying GPT 4



AN how would I use that to trim just the white space at the end of the string? ✎



To trim just the white space at the end of a string in R using the stringr package, you would use the ``str_trim()`` function with the side argument set to "right". Here's an example:

```
>_ R
```

```
library(stringr)
your_string <- "This is your string with trailing white space.  "
trimmed_string <- str_trim(your_string, side = "right")
```

In this example, ``trimmed_string`` will contain the original string but without any trailing white space.

Prompt: Querying GPT 4



AN what is the {stringr} function that will trim white space from a character string? 



The function in stringr package that will trim white space from a character string is ``str_trim()``.

Prompt: Querying GPT 4



AN how would I use that to trim just the white space at the end of the string? 



To trim just the white space at the end of a string in R using the stringr package, you would use the ``str_trim()`` function with the side argument set to "right". Here's an example:

```
>_ R
```

```
library(stringr)
your_string <- "This is your string with trailing white space.   "
trimmed_string <- str_trim(your_string, side = "right")
```

In this example, ``trimmed_string`` will contain the original string but without any trailing white space.

Prompt: Querying GPT 4

## Where We are Today

- Focus on creating guides providing relevant examples of where people might want to use this
- Presenting at small group sessions to allow people to ask questions
- Result: High adoption rates across GSK

### Development LLM Info Hub



LLMs  
Overview



LLMs  
at GSK



Prompting  
Guide



Example  
Catalogue

# Looking to Tomorrow



## A step too far?

- Could an LLM generate CDISC ADaM data?

AN How would I use R to create ADaM data? Can you give me an example of some code to generate an ADAE dataset using CDISC standards?



Creating an ADaM dataset in R involves several steps, including importing your raw data, applying the necessary transformations, and exporting the final dataset. Here's a very basic example of how you might create an ADAE (Adverse Events) dataset:

```
>_ R

# Load necessary libraries
library(dplyr)
library(tidyr)

# Import your raw data
raw_data <- read.csv("raw_data.csv")

# Assume raw_data has columns: subject_id, adverse_event, start_date, end_date

# Apply necessary transformations
adae_data <- raw_data %>%
  mutate(
    USUBJID = subject_id, # Unique Subject Identifier for each subject
    AEDECOD = adverse_event, # Standardized term for the adverse event
    AESTDTC = start_date, # Start date of the adverse event
    AEENDTC = end_date # End date of the adverse event
  ) %>%
  select(USUBJID, AEDECOD, AESTDTC, AEENDTC) # Select only the necessary columns

# Export the final dataset
write.csv(adae_data, "adae_data.csv")
```

This is a very simplified example. In practice, creating an ADaM dataset would involve more complex transformations and would need to adhere to the specific CDISC ADaM standards. You would also need to create additional variables like `AETERM` (verbatim term for the adverse event), `AESEV` (severity of the adverse event), `AESER` (whether the event is serious), and others based on your specific study protocol and statistical analysis plan.

## A step too far?

AN How would I use R to create ADaM data? Can you give me an example of some code to generate an ADAE dataset using CDISC standards? 



Creating an ADaM dataset in R involves several steps, including importing your raw data, applying the necessary transformations, and exporting the final dataset. Here's a very basic example of how you might create an ADAE (Adverse Events) dataset:

```
>_R
```

```
# Load necessary libraries
library(dplyr)
library(tidyr)
```

```
# Import your raw data
raw_data <- read.csv("raw_data.csv")
```

```
# Assume raw_data has columns: subject_id, adverse_event, start_date, end_date
```

```
# Apply necessary transformations
```

## A step too far?

- Could an LLM generate CDISC ADaM data?
- What if the model had the specifications?
- The first AI Programmer Experiment was born!

AN How would I use R to create ADaM data? Can you give me an example of some code to generate an ADAE dataset using CDISC standards?



Creating an ADaM dataset in R involves several steps, including importing your raw data, applying the necessary transformations, and exporting the final dataset. Here's a very basic example of how you might create an ADAE (Adverse Events) dataset:

```
>_ R

# Load necessary libraries
library(dplyr)
library(tidyr)

# Import your raw data
raw_data <- read.csv("raw_data.csv")

# Assume raw_data has columns: subject_id, adverse_event, start_date, end_date

# Apply necessary transformations
adae_data <- raw_data %>%
  mutate(
    USUBJID = subject_id, # Unique Subject Identifier for each subject
    AEDECOD = adverse_event, # Standardized term for the adverse event
    AESTDTC = start_date, # Start date of the adverse event
    AEENDTC = end_date # End date of the adverse event
  ) %>%
  select(USUBJID, AEDECOD, AESTDTC, AEENDTC) # Select only the necessary columns

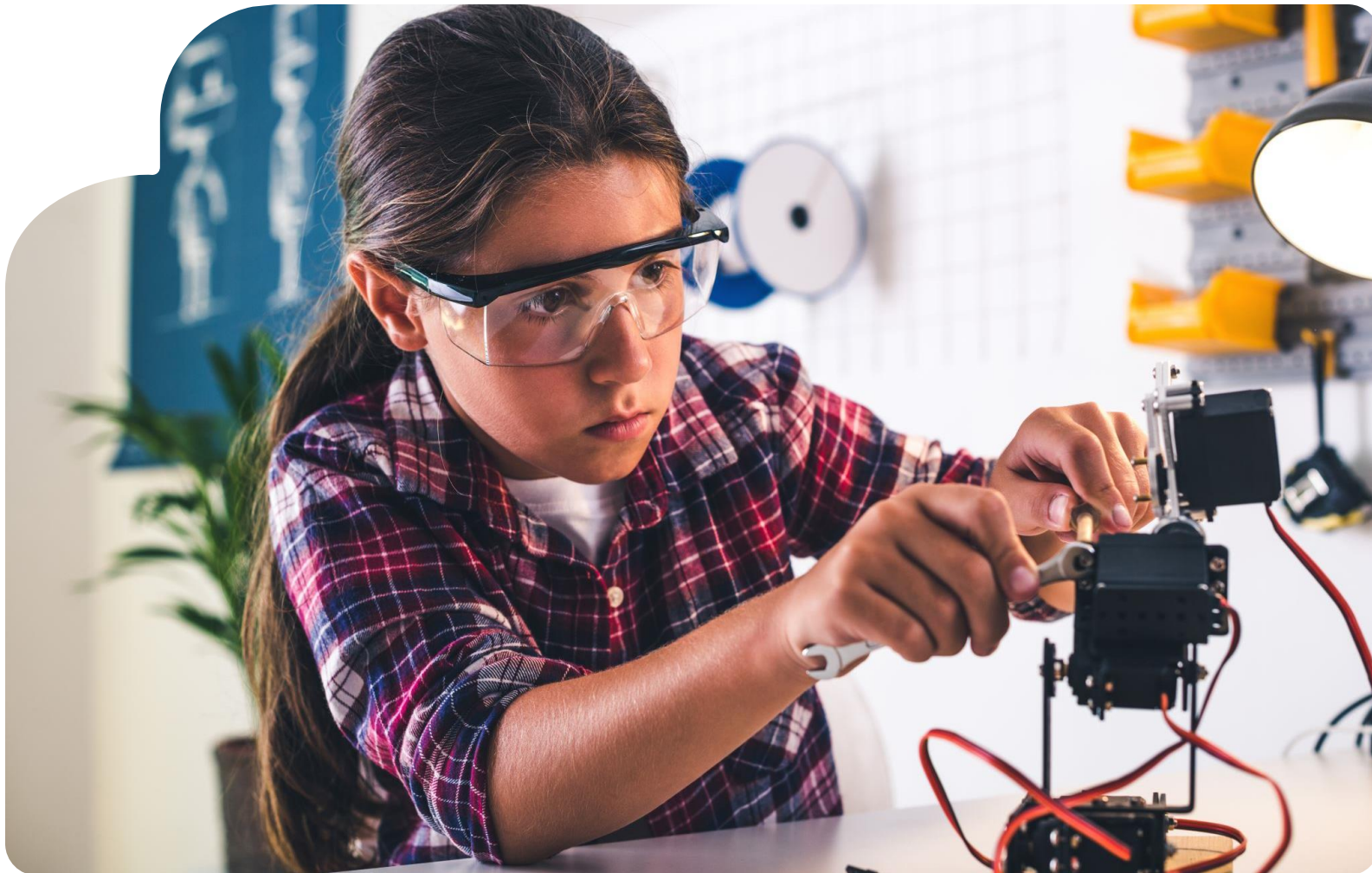
# Export the final dataset
write.csv(adae_data, "adae_data.csv")
```

This is a very simplified example. In practice, creating an ADaM dataset would involve more complex transformations and would need to adhere to the specific CDISC ADaM standards. You would also need to create additional variables like `AETERM` (verbatim term for the adverse event), `AESEV` (severity of the adverse event), `AESER` (whether the event is serious), and others based on your specific study protocol and statistical analysis plan.

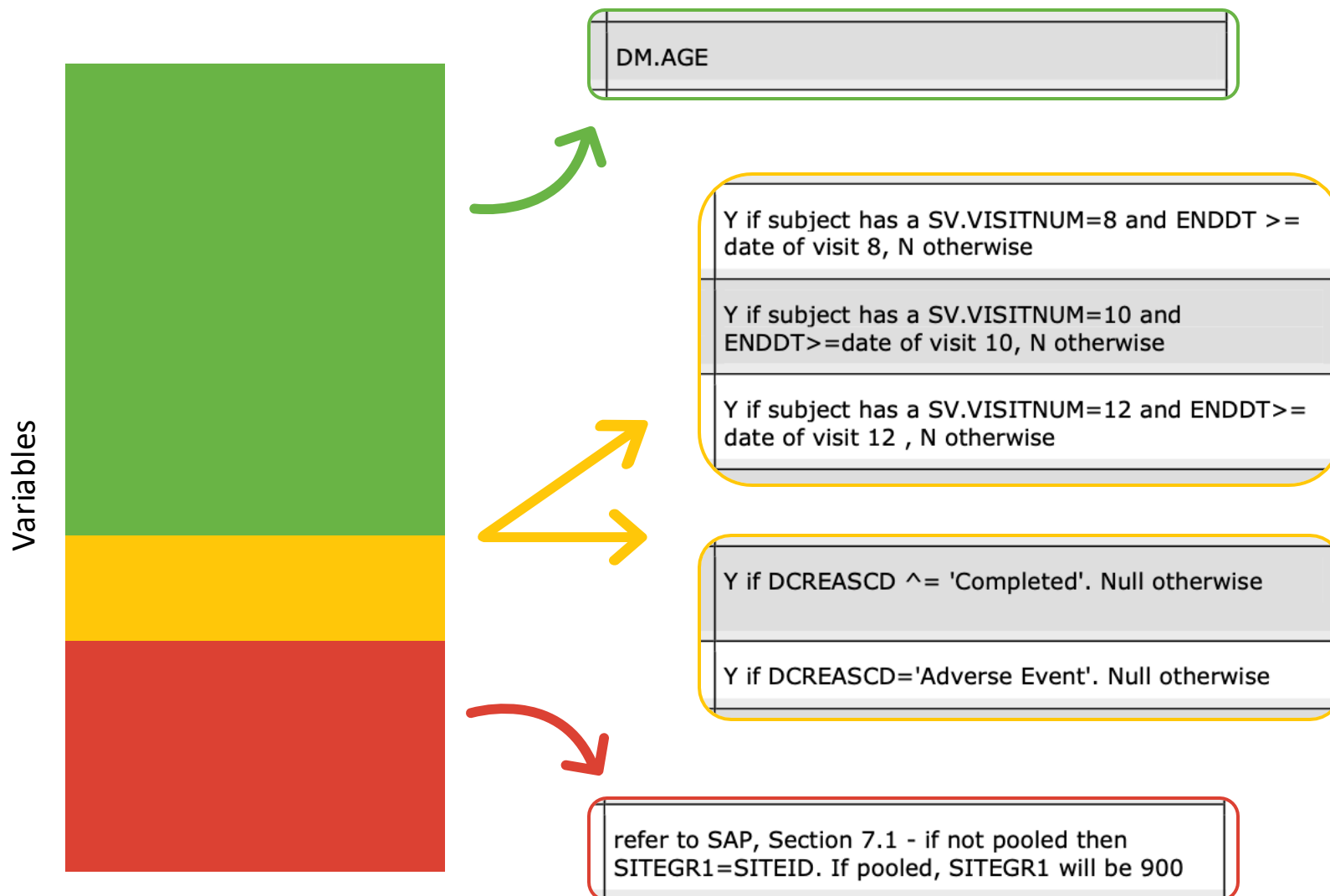
# Testing the Theory

## Initial Experimentation

- In December we worked with a contractor to test if using LLM's to read in dataset specifications
  - Kept everything external
  - Used Open CDISC datasets and external model
  - Results only in Python



# Detailed Results Overview



# Creating a Prototype

## Bringing in house

- For 3 months we worked with the contractors to build an in-house pipeline
  - Gathering sample code and specifications
  - Experimentation with fine-tuning to write pharmaverse code
  - Creating a simple pipeline
- Results were mixed



## The Project Tomorrow

- GSK entering a new phase with a view to applying to live studies
- Multi-stage prompting
- Productionisation to follow, pending internal quality benchmarks being met, including positive Programmer feedback!



## Caution

- We're not there yet!
- For every promising result, there is another disappointing one
- Ongoing experiments with Gen AI have highlighted several challenges



# Sharing Experience



## Consideration

### Bringing LLM's to the masses

- You don't need a fancy automated system to see time savings
- Give people the opportunity to ask questions
- Guidance can help
- Best training is experience



ON AIR

## Considerations

Tech partnership is needed

- LLM's require a whole new set of infrastructure



## Considerations

Guides are helpful

- The world of LLM's is different from the world of clinical programming



## Considerations

Finetuning probably isn't the answer

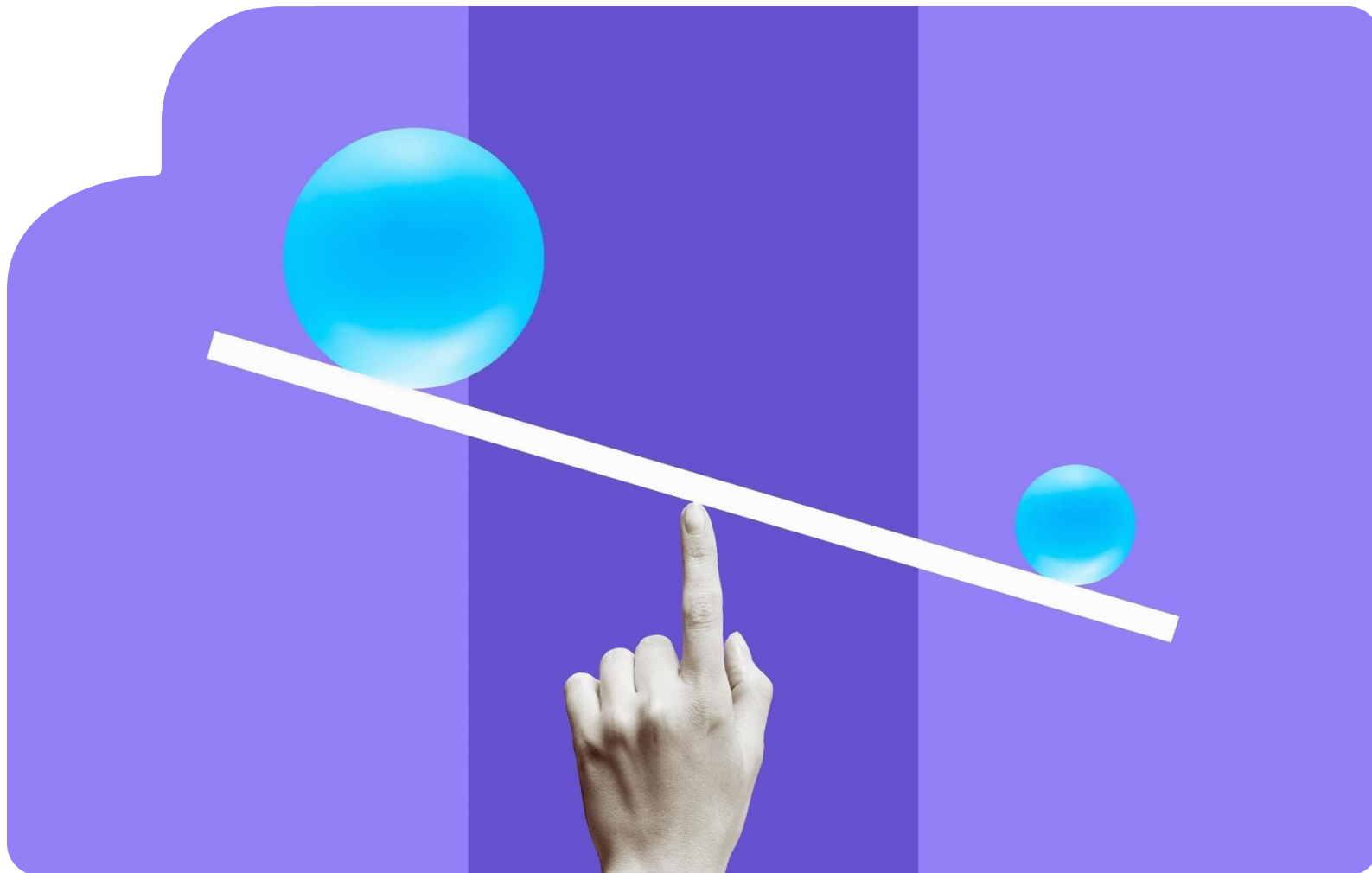
- While the easiest to understand conceptually, finetuning probably won't help you
- Especially if you are doing this to support an R transition



# Considerations

## Trade-offs

- When making a pipeline, there are trade-offs to be made with respect to pipeline simplicity and prompt complexity



# Considerations

Context is important

- Dataset specifications are a good place to start LLM journey's
  - There is actually relatively little outside context needed
- We don't actually make that many documents



# Thinking Beyond Tomorrow



# What does the future hold

- Our current jobs will change
  - From code writers to readers/reviewers
  - Prompt writing



# Questions



# Reference

- "ChatGPT Could Replace Telemarketers, Teachers, and Traders." The Economist, The Economist Newspaper, 14 April 2023, [www.economist.com/graphic-detail/2023/04/14/chatgpt-could-replace-telemarketers-teachers-and-traders](https://www.economist.com/graphic-detail/2023/04/14/chatgpt-could-replace-telemarketers-teachers-and-traders).
- Hu, Krystal. "ChatGPT Sets Record for Fastest Growing User Base: Analyst Note." Reuters, Thomson Reuters, 1 February 2023, [www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/](https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/).

