

Comparison of Generative AI and Manual Data Programming in a Lupus Health Productivity Loss Study

OBJECTIVE

- To evaluate the performance of a generative artificial intelligence (GenAI) assistant in analyzing health productivity losses in a U.S. claims database, compared with analysis using human-written code

CONCLUSIONS

- ChatGPT-4 can replicate simple data-related tasks, such as patient selection, when the input is broken down into separate tasks, with an acceptable number of prompt iterations
- For the coding of complex tasks, human intervention and high-level programming skill remain necessary to leverage ChatGPT’s capabilities
- The potential of ChatGPT in health economics has yet to be fully realized and the utility demonstrated in this study warrants further investigation



→ Scan QR code for copy of poster
International Society for Pharmacoeconomics and Outcomes Research (ISPOR) 2025
May 13–16; Montreal, Canada

Introduction

- In recent years, there has been growing adoption of generative artificial intelligence (GenAI); however, its application in health economics has not been widely explored¹
- Outside of health economics, the utility of ChatGPT (a GenAI assistant) has been recognized in tasks related to code generation²
- This study evaluated the coding performance of ChatGPT to analyze real-world data on health-related productivity losses in a U.S. commercially insured population compared with existing analysis undertaken by human coders (see ISPOR 2025 poster **EE137**)

Methods

Research goal

- To understand the health-related productivity losses associated with patients newly diagnosed with systemic lupus erythematosus (SLE), using a real-world dataset

Study cohort

- Data were obtained from the IBM® MarketScan® database, the IBM® MarketScan® Health Productivity and Management (HPM) database, and Medicare claims, covering the period from January 1, 2016, to December 31, 2022
- Two adult cohorts were defined: newly diagnosed SLE and non-SLE (which included other non-SLE conditions)
- Newly diagnosed patients with SLE were defined as having ≥2 outpatient claims with an International Classification of Diseases (ICD)-9/10 code for SLE and ≥30 days between claims, or ≥1 inpatient claim with an ICD-9/10 code for SLE
 - The ICD-9/10 codes used for patient selection included: 7100, M32, M321, M3210, M3211, M3212, M3213, M3214, M3215, M3219, M328, and M329
 - The index date was the first SLE diagnosis within the study period
 - Patients diagnosed with SLE 12 months or less prior to the index date and those with drug-induced SLE were excluded
- Non-SLE was defined as having no SLE claim/diagnosis during the study period and eligible for disability benefits
 - The index dates were randomly selected from the available range and randomly assigned (seed number: 100) to patients to simulate the distribution of index dates within the SLE cohort
- Wage rates and benefit data were extracted from the Bureau of Labor Statistics³

Statistical analyses

- Propensity score weighting was used to balance baseline differences between newly diagnosed SLE and non-SLE cohorts
- An inverse probability of treatment weighting (IPTW) cross-sectional linear regression evaluated the health productivity losses associated with SLE, after controlling for patient demographics, index year, U.S. region, and covariates

ChatGPT coding process

- The ability of ChatGPT to replicate manual analyses of productivity losses was assessed
- The artificial intelligence (AI) coding replication process was evaluated in four steps:
 - Researchers** completed all tasks using Structured Query Language (SQL) and R, including coding and visualization of results
 - Human-written code was divided into tasks**, with corresponding prompts created for ChatGPT-4
 - ChatGPT-generated code** was tested against the original human-generated results
 - Human intervention** was introduced if ChatGPT-4 was unable to generate the correct code to complete the task after 10 prompt attempts
- Figure 1** presents an example of prompt and ChatGPT response for the extraction of yearly absenteeism records from the HPM database
- The quality of written prompts has an impact on the AI output and requires careful consideration
- Criteria to evaluate the coding performance of ChatGPT included:
 - Success:** Measure of whether ChatGPT was able to generate the requested code
 - Efficiency (SQL):** Measure of the number of tables / temporary views generated per task
 - Efficiency (R):** Measure of the number of commands used per task
 - Accuracy:** Measure of whether ChatGPT was able to replicate the correct results

Figure 1. Example human prompt and ChatGPT output for data extraction task

Attempt 1	
Prompt	ChatGPT output
Good, now we will move forward and generate a new temp view named abs_2016_gpt, by select all variables from cora.conti_members_gpt and \${d.hpm}.abs2016_v3, add a new variable 2016 as year, and another new variable 2016 as abs_year. Join by id, note that patid is from cora.conti_members_gpt and enrolid is from \${d.hpm}.abs2016_v3.	CREATE OR REPLACE TEMP VIEW abs_2016_gpt AS SELECT cm.*, -- Selects all columns from 'cora.conti_members_gpt' abs.*, -- Selects all columns from '\${d.hpm}.abs2016_v3' 2016 AS year, -- Adds a new column 'year' with value 2016 2016 AS abs_year -- Adds a new column 'abs_year' with value 2016 FROM cora.conti_members_gpt cm LEFT JOIN \${d.hpm}.abs2016_v3 abs ON cm.patid = abs.enrolid;

Results

- Table 1** presents ChatGPT’s programming performance by coding outcome
- The human-written code was divided into 75 separate tasks and ChatGPT generated code for each task
 - Of these tasks, 77.3% were completed without the need for revisions
 - 18.7% of tasks required <10 prompt revisions to achieve accurate results
 - The remaining 4% of tasks needed human intervention (e.g., calculation of Charlson Comorbidity Index scores using ICD-9/10 coding)

Overall, ChatGPT:

- Demonstrated excellent performance when conducting date- and time-related tasks
- Experienced difficulties when running requests for high-complexity tasks and when facing an abundance of options

Table 1. ChatGPT programming performance evaluation

Part 2: Charlson comorbidity scores calculation		Outcomes			Number of revisions
		Success (Yes/No)	Efficiency (continuous)*	Accuracy (Yes/No)	
Step 1	Warming up – input ChatGPT-generated code to refresh and catch up on the work we were doing	Yes	N/A	Yes	N/A
Step 2	Select all the disease records for eligible SLE patients	Yes	1	Yes	2
Step 3	Limiting the disease records to the pre-index date	Yes	1	Yes	1
Step 4	Generate indicator variables for diseases that constituted CCI score	No	1	Yes (with human intervention)	10
Step 5	Assign the score weight to each of the indicator variables	Yes	1	Yes	2
Step 6	Drop duplicates of the disease records	Yes	1	Yes	0
Step 7	Calculate the CCI score for each eligible SLE patient (the final table for CCI index score calculation SLE-CCI set)	Yes	1	Yes	0

*Efficiency is a measure of the number of tables / temporary views / commands per task; a low value indicates high efficiency
CCI, Charlson Comorbidity Index; NA, not applicable; SLE, systemic lupus erythematosus

References 1. Parray AA, et al. Glob Transit 2023;5:50–54; 2. Huang Y, et al. J Glob Health 2024;14:04070; 3. U.S. Bureau of Labor Statistics. Disability insurance plans: trends in employee access and employer costs. Available at: <https://www.bls.gov/opub/btn/volume-4/disability-insurance-plans.htm> (Accessed April 25, 2025). **Disclosures** TT, CM, and FZ are employees of Biogen and may hold Biogen stock. **Funding** This study was funded by Biogen (Cambridge, MA, USA). Editorial support was provided by Selene Medical Communications (Macclesfield, UK), funded by Biogen. Copyright ©2025 Biogen Inc. All rights reserved.