

## Tutorial



# Data science through natural language with ChatGPT's Code Interpreter

Sangzin Ahn <sup>1,2,\*</sup>

<sup>1</sup>Department of Pharmacology and Pharmacogenomics Research Center, Inje University College of Medicine, Busan 47392, Korea

<sup>2</sup>Center for Personalized Precision Medicine of Tuberculosis, Inje University College of Medicine, Busan 47392, Korea

## OPEN ACCESS

Received: Mar 25, 2024

Revised: May 2, 2024

Accepted: May 26, 2024

Published online: May 29, 2024

### \*Correspondence to

Sangzin Ahn

Department of Pharmacology and  
Pharmacogenomics Research Center,  
Center for Personalized Precision Medicine  
of Tuberculosis, Inje University College of  
Medicine, 75 Bokji-ro, Busanjin-gu, Busan  
47392, Korea.  
Email: sangzinahn@inje.ac.kr

Copyright © 2024 Translational and Clinical  
Pharmacology

It is identical to the Creative Commons  
Attribution Non-Commercial License ([https://  
creativecommons.org/licenses/by-nc/4.0/](https://creativecommons.org/licenses/by-nc/4.0/)).

### ORCID iDs

Sangzin Ahn 

<https://orcid.org/0000-0003-2749-0014>

### Funding

This work was supported by the National  
Research Foundation of Korea (NRF) grant  
funded by the Korean government (MSIT)  
(No. 2018R1A5A2021242).

### Conflict of Interest

- Authors: Nothing to declare
- Reviewers: Nothing to declare
- Editors: Nothing to declare

### Usage of AI Tools

The AI chatbot Claude 3 Opus was used in  
the process of writing and revising the outline  
of the manuscript, as well as in the process  
of revising the wording and grammar of the  
manuscript.

## ABSTRACT

Large language models (LLMs) have emerged as a powerful tool for biomedical researchers, demonstrating remarkable capabilities in understanding and generating human-like text. ChatGPT with its Code Interpreter functionality, an LLM connected with the ability to write and execute code, streamlines data analysis workflows by enabling natural language interactions. Using materials from a previously published tutorial, similar analyses can be performed through conversational interactions with the chatbot, covering data loading and exploration, model development and comparison, permutation feature importance, partial dependence plots, and additional analyses and recommendations. The findings highlight the significant potential of LLMs in assisting researchers with data analysis tasks, allowing them to focus on higher-level aspects of their work. However, there are limitations and potential concerns associated with the use of LLMs, such as the importance of critical thinking, privacy, security, and equitable access to these tools. As LLMs continue to improve and integrate with available tools, data science may experience a transformation similar to the shift from manual to automatic transmission in driving. The advancements in LLMs call for considering the future directions of data science and its education, ensuring that the benefits of these powerful tools are utilized with proper human supervision and responsibility.

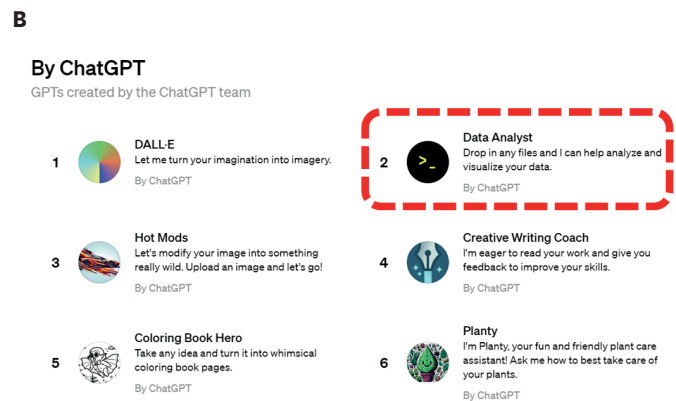
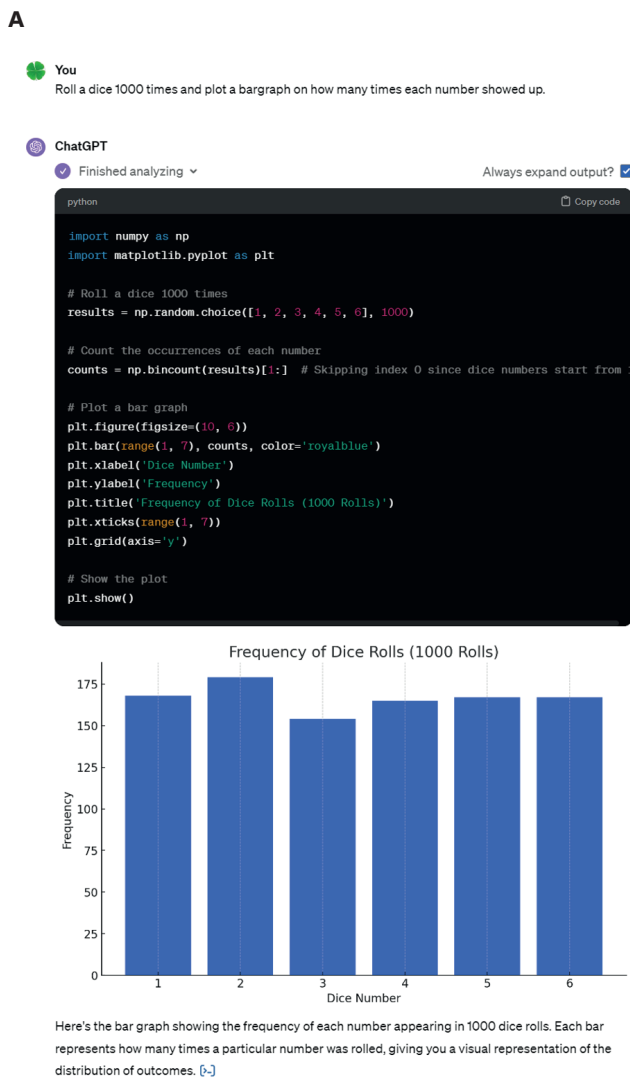
**Keywords:** Artificial Intelligence; Data Science; Data Analysis; Natural Language Processing; Machine Learning

## INTRODUCTION

Computer programming has become an indispensable tool for biomedical researchers to carry out essential research tasks [1]. In recent years, large language models (LLMs) have demonstrated remarkable capabilities in understanding and generating human-like text [2]. A global survey conducted by Nature in July 2023 found that about one-third of postdoc respondents reported using AI chatbots for various tasks, with refining text (63%) and code-related tasks (56%) being the most common applications [3]. This high prevalence highlights the significant impact LLMs are having on the research landscape.

A notable development in this field is the introduction of ChatGPT Code Interpreter (CI), a new feature launched on July 6, 2023. This powerful tool allows ChatGPT to run code and interact with uploaded files, enabling users to analyze data, create charts, edit files, perform math, and more (<https://openai.com/blog/chatgpt-plugins#code-interpreter>). CI can be invoked simply by prompting ChatGPT with a command that requires computation, such as “Roll a dice 1000 times and plot a bar graph on how many times each number showed up” (**Fig. 1A**). Additionally, users can access a GPT (customized chatbots for specific tasks) named Data Analyst, found in the GPTs by ChatGPT section, which is specifically designed for utilizing CI on data analysis tasks (**Fig. 1B**).

While CI was initially available only to paid users, as of May 30, 2023, free users can also access this functionality, albeit with usage restrictions. Similarly, Julius AI (<https://julius.ai>), another LLM-based tool, offers data analysis capabilities but also imposes usage limitations for free users. Despite these constraints, the increasing accessibility of LLM-based data



**Figure 1.** Demonstration of ChatGPT Code Interpreter functionality. (A) An example prompt and output for rolling a dice 1,000 times and generating a bar plot. (B) The Data Analyst GPT, a specialized chatbot for data analysis tasks, found in the GPTs by ChatGPT section.

analysis tools, and the anticipated development of open-sourced alternatives, such as OpenCodeInterpreter [4], are expected to empower a growing number of researchers to leverage LLMs for streamlining their data analysis workflows and gaining valuable insights.

This tutorial aims to demonstrate the potential of LLMs with code writing and execution abilities as a powerful tool for researchers, enabling them to conduct data analysis using natural language and receive assistance in code generation. By using materials from a tutorial published in Translational and Clinical Pharmacology in December 2022 [5], this tutorial will showcase how similar analyses can be performed through natural language interactions with an LLM chatbot. It is hoped that this approach represents an exciting and promising development in the field of biomedical research.

## DATASET

A modified version of the International Warfarin Pharmacogenetics Consortium open access dataset (<https://www.pharmgkb.org/downloads>), the identical dataset from the previous tutorial, was used in this article [6]. The original dataset was preprocessed by removing patients with zero warfarin dose, unstable dosing, or missing critical demographic and genetic data. Ages were assigned within defined 10-year bins, and missing values were imputed using the MissForest algorithm. Categorical variables were converted to discrete numerical values to streamline analyses. Details of the preprocessing steps are described in the previous tutorial and the final preprocessed dataset is available for download at [https://github.com/mahlernim/warfarin\\_prediction\\_kscpt\\_tutorial](https://github.com/mahlernim/warfarin_prediction_kscpt_tutorial) [5].

## DATA LOADING AND EXPLORATION

Most major file types used in data science, such as csv, xlsx, txt, pickle, tsv, json, yaml can be uploaded and be utilized by CI. A detailed description of the coding of the dataset can drastically help CI to understand the dataset and perform downstream analytical tasks using the context information. In this demonstration, the “Dataset and preprocessing” section from the previous tutorial article that contains information of the coding of the dataset was given to the chatbot, and was asked to explore and come up with a research proposal. The entire chat record with Data Analyst is available in **Supplementary Data 1**. In the background, the chatbot utilized its CI tool to load the csv file and read the column names and first 5 rows and get a grasp about the dataset (**Fig. 2A**). Then it will combine all given information and propose a research plan, including background, objectives, methodology, expected outcomes and significance (**Fig. 2B**). This type of approach where the researcher provides the dataset and information about the dataset and asking for further steps can be particularly beneficial for junior scientists to cultivate and refine their research planning skills. Another recommended prompt is “explore the dataset and give me insights for further analyses,” which initiates to perform basic analyses including basic descriptive analytics, correlation, regression and scatter plots. For the sake of brevity, this tutorial will directly move on to model development.

A

You

IWPC\_cleaned.csv  
Spreadsheet

The following is the description about the dataset, which is from a previously published paper.  
....  
"Dataset and preprocessing" section pasted here  
....  
Briefly explore the dataset and write up a short research proposal.

Data Analyst

Finished analyzing ▾ Always expand output? ✓

```
python
# First, let's load the dataset to understand its structure and contents.
import pandas as pd

# Load the dataset
file_path = '/mnt/data/IWPC_cleaned.csv'
data = pd.read_csv(file_path)

# Display the first few rows of the dataframe to understand its structure
data.head()
```

Result

	Gender	Age	Race	Height	Weight	Diabetes	CHF	ValveReplacement	\
0	1	83	1	172.9994	69.0	0	0	0	
1	0	56	2	161.2900	92.5	0	0	0	
2	1	48	1	183.0070	113.0	1	1	0	
3	0	70	1	149.9870	52.0	0	1	0	
4	0	52	2	172.7200	135.0	1	0	0	

	Aspirin	Tylenol	Simvastatin	Amiodarone	Vitamin	Smoking	CYP2C9	\
0	1	1	0	0	0	0	0	
1	0	0	0	0	0	0	1	
2	0	0	0	0	0	0	0	
3	1	0	0	0	0	0	0	
4	0	0	1	0	0	0	0	

	VKORC1	Warfarin
0	2	157.5
1	2	126.0
2	2	122.5
3	2	122.5
4	1	121.9

The dataset consists of various demographic, pharmacogenomic, and medication information for patients who have been prescribed warfarin, along with their stable weekly warfarin dose. The columns include:

Omitted for brevity

B

**Research Proposal**

Title:

"Optimizing Warfarin Dosing in Diverse Populations: A Machine Learning Approach"

Background:

Warfarin is a widely used anticoagulant for preventing thrombosis and thromboembolism. However, its dosing is challenging due to its narrow therapeutic index and the wide variation in dose requirements among individuals, influenced by genetic makeup, demographic factors, and interaction with other medications. The IWPC dataset, which includes demographic, pharmacogenomic, and medication information, provides a unique opportunity to explore these influences in depth.

Objectives:

1. To identify the most significant factors influencing warfarin dose requirements using machine learning feature selection techniques.
2. To develop a predictive model for optimal warfarin dosing that incorporates genetic markers, demographic factors, and co-medications.
3. To evaluate the model's performance across different racial groups to ensure its applicability in diverse populations.

Methodology:

- **Data Analysis:** Initial exploration of the dataset to understand the distribution of various features and the target variable (warfarin dose). This includes statistical analyses and visualization.
- **Feature Selection:** Utilization of machine learning techniques such as random forests and gradient boosting to identify the most significant predictors of warfarin dose.
- **Model Development:** Training of several machine learning models, including linear regression, decision trees, random forests, and gradient boosting, to predict the optimal warfarin dose. Model selection will be based on performance metrics such as  $R^2$ , RMSE (Root Mean Square Error), and MAE (Mean Absolute Error).
- **Model Evaluation:** Cross-validation and external validation using a separate dataset (if available) to assess the model's generalizability and performance across different racial groups.

Expected Outcomes:

- A comprehensive list of the most impactful predictors of warfarin dose requirements.
- A robust predictive model that can accurately determine the optimal warfarin dose for individual patients, potentially improving treatment outcomes and reducing the risk of adverse effects.
- Insights into the model's performance and applicability in diverse populations, which could guide personalized medicine approaches in anticoagulation therapy.

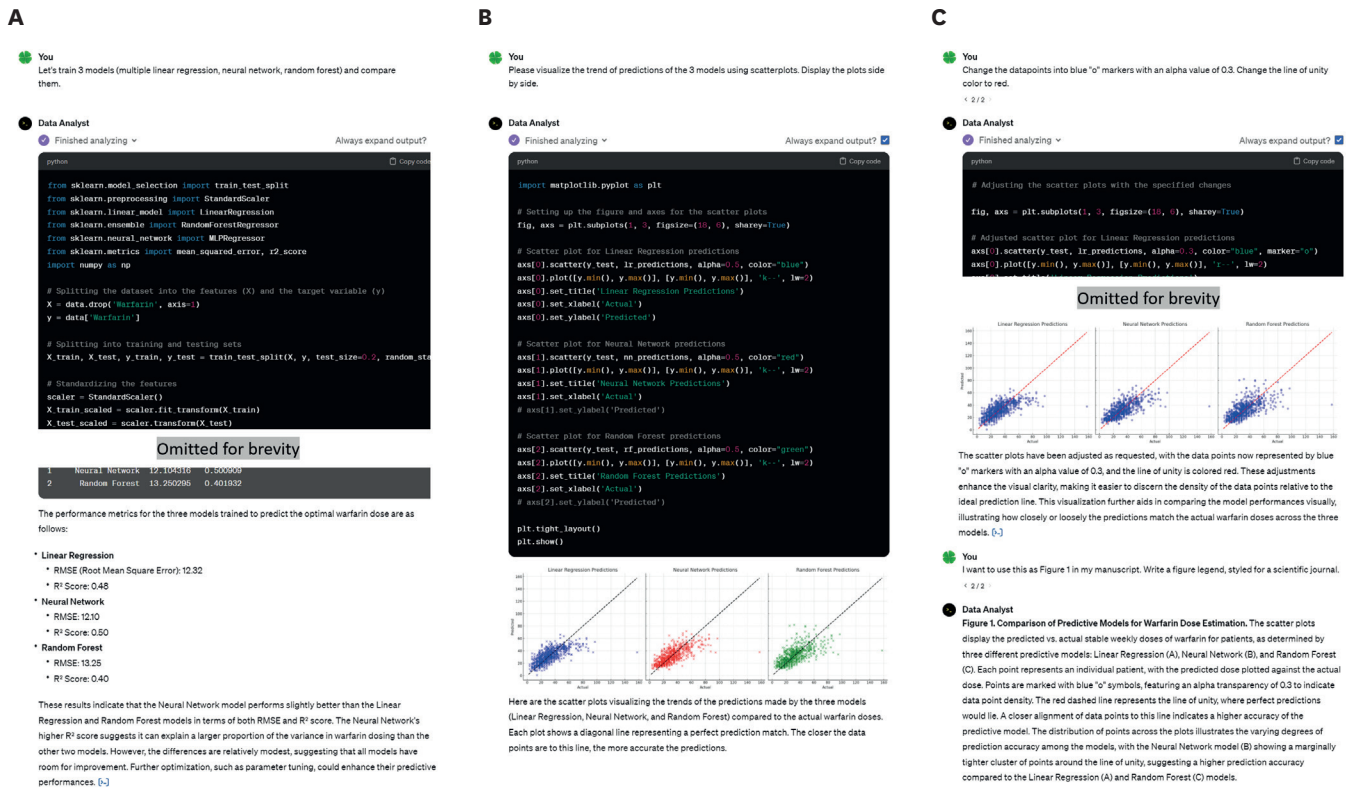
Significance:

This research could significantly impact clinical practice by providing a tool for personalized warfarin dosing, reducing the risk of under- or overdosing, and thereby decreasing the incidence of adverse events and improving patient outcomes. Additionally, the findings could contribute to a better understanding of the role of genetics and other factors in drug metabolism and efficacy, informing future pharmacogenomic studies and personalized medicine approaches. [3]

**Figure 2.** Utilization of Code Interpreter to explore a dataset and generate a research proposal. (A) Prompting ChatGPT to explore an uploaded dataset. ChatGPT uses Code Interpreter to load a dataset, display the first few rows, and understand its structure and contents. (B) A research proposal generated by ChatGPT, including the title, background, objectives, methodology, expected outcomes, and significance, based on the dataset and provided description.

**MODEL DEVELOPMENT AND COMPARISON**

Following the methodology outlined in the previous tutorial, three models were trained using the preprocessed dataset: Multiple Linear Regression, Neural Network, and Random Forest. Without being provided specific metrics for comparison, the chatbot utilized root mean square error and  $R^2$  scores to evaluate the models' performance (Fig. 3A). Subsequently, the chatbot was instructed to generate scatterplots for visualizing the performance of the three models (Fig. 3B). It is worth noting that during the visualization process, CI may employ creative visualization settings that might not always align with the user's preferences. In such cases, the user can request specific changes to the settings (Fig. 3C). When a plot is intended



**Figure 3.** Model development, comparison, and visualization using Code Interpreter. (A) ChatGPT was prompted to train and compare the performance of Linear Regression, Neural Network, and Random Forest models. Code Interpreter calculated root mean square error and R<sup>2</sup> scores as evaluation metrics. (B) Visualization of the three predictive models using scatterplots, displaying the predicted to actual warfarin dose values. (C) Adjustments to the scatterplot settings and a request for a figure legend.

to be used as a figure in the manuscript, the user can ask the chatbot to draft a figure legend (**Fig. 3C**). Leveraging its access to the dataset and previous interactions with the user, the chatbot generates a legend that provides appropriate context. This draft can serve as a foundation for the user to refine further during subsequent manuscript preparation stages.

## PERMUTATION FEATURE IMPORTANCE

To further analyze the data, the user requested, “I want to visualize the relationship between each feature and the dependent variable, and also show what the important features are for each model.” This interaction illustrates a scenario where the user may not be familiar with the specific methodology of permutation feature importance but understands its significance and relevance in the analysis. The chatbot interprets the user’s intent and conducts permutation feature importance, providing visualizations of the results (**Fig. 4A**). In this demonstration, the user provides specific adjustment requirements, which the chatbot implements by modifying the Python code accordingly (**Fig. 4B**).

## PARTIAL DEPENDENCE PLOTS

Subsequently, the chatbot was instructed to generate partial dependence plots for each model. When provided with concise instructions, the chatbot makes several assumptions



**A**

**You**  
I want to visualize the relationship between each feature and the dependent variable, and also show what the important features are for each model.

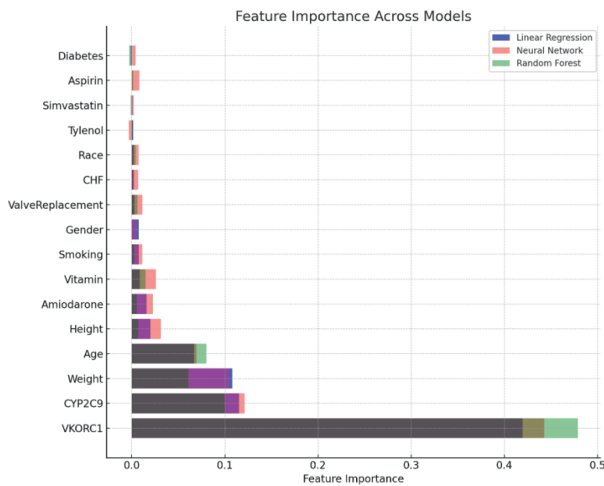
**Data Analyst**  
Finished analyzing ✓ Always expand output? ✓

```
python
from sklearn.inspection import permutation_importance

# Calculating permutation importance for each model to identify feature importance
lr_importance = permutation_importance(lr_model, X_test_scaled, y_test, n_repeats=30)
nn_importance = permutation_importance(nn_model, X_test_scaled, y_test, n_repeats=30)
rf_importance = permutation_importance(rf_model, X_test_scaled, y_test, n_repeats=30)

# Aggregating feature importance
feature_importance_df = pd.DataFrame({
    'Feature': X.columns,
    'Linear Regression Importance': lr_importance.importances_mean,
    'Neural Network Importance': nn_importance.importances_mean,
    'Random Forest Importance': rf_importance.importances_mean
}).sort_values(by='Linear Regression Importance', ascending=False)
```

Omitted for brevity



This visualization presents the feature importance as determined by three different models: Linear Regression, Neural Network, and Random Forest. Each bar represents a feature from the dataset, and

Omitted for brevity

**B**

**You**  
Make the following adjustments  
- Use boxplots instead.  
- Show separate plots for each model side by side instead of superimposing.  
- Order the Y-axis from highest importance to lowest importance.  
- Use same colors for all models

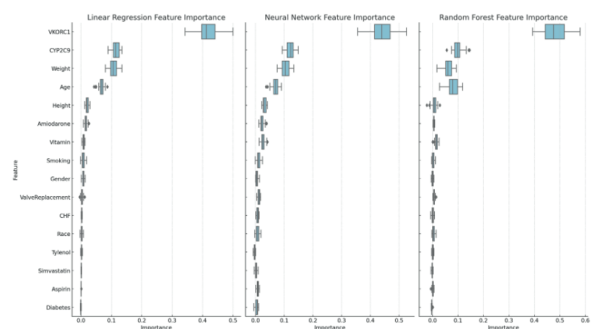
**Data Analyst**  
Finished analyzing ✓ Always expand output? ✓

```
python
# Preparing data for boxplots of feature importance for each model
lr_importances_sorted = pd.DataFrame(lr_importance.importances).T
lr_importances_sorted.columns = X.columns
lr_importances_sorted = lr_importances_sorted[feature_importance_df['Feature']]

nn_importances_sorted = pd.DataFrame(nn_importance.importances).T
nn_importances_sorted.columns = X.columns
nn_importances_sorted = nn_importances_sorted[feature_importance_df['Feature']]

rf_importances_sorted = pd.DataFrame(rf_importance.importances).T
rf_importances_sorted.columns = X.columns
```

Omitted for brevity



The boxplots display the distribution of feature importance scores across three predictive models for warfarin dose estimation: Linear Regression, Neural Network, and Random Forest. Each plot corresponds to one model, with the features ordered from highest to lowest importance based on the Linear Regression model's importance scores. This ordering provides a consistent basis for comparison across models.

The color "skyblue" is used uniformly across all models to maintain visual consistency. The boxplots allow us to observe not only the central tendency of the importance scores for each feature but also the variability in these scores across multiple permutations. This visualization aids in identifying which features consistently have a strong influence on the model predictions across different modeling approaches, highlighting their potential relevance in warfarin dosing. [6]

**Figure 4.** Permutation feature importance analysis using Code Interpreter. (A) User request for visualizing the relationship between features and the dependent variable, and identifying important features for each model. ChatGPT performs permutation feature importance and provides code for aggregating and visualizing the results. (B) User-specified adjustments made to the code by requesting in natural language.

and attempts to fulfill the user's request. In this demonstration, the chatbot utilizes the three most influential features identified in the previous section and creates plots for each feature (Fig. 5A). As with previous sections, users can request additional adjustments using natural language, enabling them to obtain the desired results (Fig. 5B).

## ADDITIONAL ANALYSES AND RECOMMENDATIONS

Users can simply ask, "What other analyses would you recommend?" to receive advice from the chatbot (Fig. 6A). The chatbot's recommendations are of high quality, with many

**A**

**You**  
Next, I want to draw partial dependence plots for each model.

```
python
from sklearn.inspection import plot_partial_dependence

# Selecting top 3 important features for simplicity in visualization
top_features = feature_importance_df['Feature'].head(3)

# Setting up the figure for partial dependence plots
fig, axs = plt.subplots(len(top_features), 3, figsize=(18, 6), sharex='col', sharey='row')

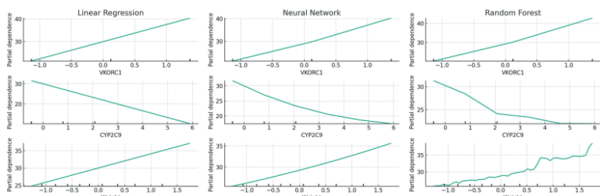
# Generating partial dependence plots for each of the top 3 features across the 3 models
for i, feature in enumerate(top_features):
    feature_idx = X.columns.get_loc(feature)

    # Linear Regression
    plot_partial_dependence(lr_model, X_train_scaled, features=[feature_idx], ax=axs[i, 0])
    if i == 0:
        axs[i, 0].set_title('Linear Regression')

    # Neural Network
    plot_partial_dependence(nn_model, X_train_scaled, features=[feature_idx], ax=axs[i, 1])

    # Random Forest
    plot_partial_dependence(rf_model, X_train_scaled, features=[feature_idx], ax=axs[i, 2])
```

Omitted for brevity



The partial dependence plots for the top three features, based on their importance in the Linear Regression model, have been generated across the three models: Linear Regression, Neural Network, and Random Forest. Each row in the figure corresponds to one of the top features, and each column represents one of the models.

These plots illustrate how the predicted warfarin dose varies with changes in a single feature while holding all other features constant at their average values. This visualization helps in understanding the relationship between each feature and the warfarin dose in the context of each model, providing insights into how different features influence the prediction of warfarin dosing.

- The **Linear Regression** column shows linear relationships, as expected, given the nature of the model.
- The **Neural Network** and **Random Forest** columns may show more complex, non-linear relationships, reflecting these models' ability to capture more intricate patterns in the data.

These findings can help guide clinical decisions by highlighting the features that have the most significant impact on warfarin dosing according to different predictive models. [3]

**B**

**You**  
Make the following arrangements  
- Make PDPs for each model using top 6 features.  
- Transpose the arrangement. Each row should be different models.

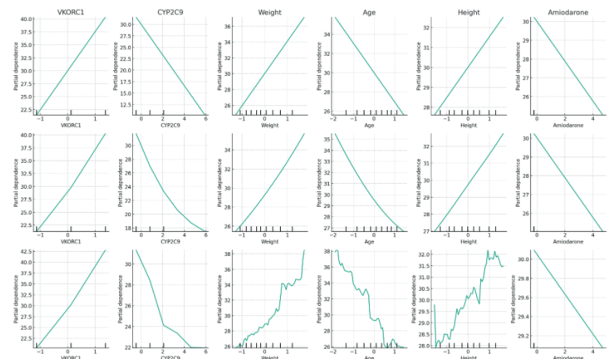
```
python
from sklearn.inspection import PartialDependenceDisplay

# Selecting top 6 important features for extended visualization
top_6_features = feature_importance_df['Feature'].head(6).tolist()

# Setting up the figure for partial dependence plots with the new arrangement
fig, axs = plt.subplots(3, len(top_6_features), figsize=(20, 12))

for i, model in enumerate([lr_model, nn_model, rf_model]):
    for j, feature in enumerate(top_6_features):
        feature_idx = X.columns.get_loc(feature)
        disp = PartialDependenceDisplay.from_estimator(
            model,
            X_train_scaled,
            features=[feature_idx],
            feature_names=X.columns,
            ax=axs[i, j],
            grid_resolution=50
        )
        if i == 0:
            axs[i, j].set_title(feature)
        if j == 0:
            axs[i, 0].set_ylabel(f"{'Linear Regression', 'Neural Network', 'Random Forest'}")

plt.tight_layout()
plt.show()
```



Omitted for brevity

**Figure 5.** Partial dependence plots generated using Code Interpreter. (A) User request for partial dependence plots for each model. ChatGPT generates plots for the top 3 features across each model, illustrating how the predicted warfarin dose varies with changes in a single feature while holding all other features constant at their average values. (B) User-specified adjustments made to the code by requesting in natural language.

of them being essential for real-world studies. Another recommended prompt to explore the creativity and ability of the chatbot is to ask it to devise the most creative analysis it can perform (**Fig. 6B**). In this demonstration, the chatbot conducted principal component analysis and k-means clustering for patient subgroup identification, generated a heat map of average warfarin dose by age and gender, and created a bar chart of average warfarin dose by medication combinations. Due to the inherent randomness of LLMs, the results may vary each time, even when provided with identical prompts.

**A**

**You**  
What other analyses would you recommend?

**Data Analyst**  
Given the work done so far and aiming to further enrich the insights from the dataset and predictive models, I recommend the following additional analyses:

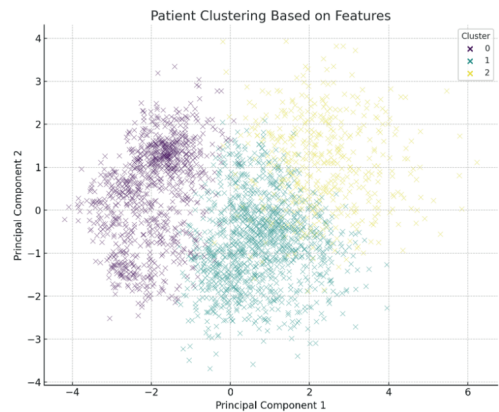
- Hyperparameter Tuning:** Each model could potentially improve in performance through hyperparameter optimization. Techniques such as grid search or randomized search can be used to find the best set of hyperparameters for the Neural Network and Random Forest models.
- Model Interpretability and Explanation:** Beyond partial dependence plots, other techniques like SHAP (SHapley Additive exPlanations) values could provide deeper insights into how each feature contributes to individual predictions. This can be especially useful for complex models like Neural Networks and Random Forests.
- Cross-Validation:** Implementing k-fold cross-validation can help assess the models' stability and performance across different subsets of the data, reducing the potential bias that might come from a single train-test split.
- Interaction Effects:** Investigating interaction effects between features can reveal complex relationships that might affect warfarin dosing. Some machine learning models can capture these interactions implicitly, but understanding and modeling them explicitly could provide additional clinical insights.
- External Validation:** If an external dataset is available, validating the models on this separate dataset can test their generalizability and robustness in different populations or settings.
- Clinical Outcome Prediction:** Besides predicting the warfarin dose, it could be valuable to predict clinical outcomes such as the risk of bleeding or thromboembolism based on the prescribed dose and patient characteristics. This approach would link dosing directly to patient outcomes, providing a more comprehensive view of treatment efficacy.
- Cost-effectiveness Analysis:** Assessing the cost-effectiveness of using these predictive models in clinical practice could provide important insights into their potential economic impact. This analysis could consider factors like reductions in adverse events, hospitalizations, and monitoring costs.
- Incorporation of Additional Data Sources:** If more patient information becomes available, incorporating additional variables (e.g., dietary habits, alcohol consumption, or other medication use) could potentially improve model predictions by capturing more factors that influence warfarin metabolism and effectiveness.
- Comparative Effectiveness Research:** For patients with alternative anticoagulation options, comparative effectiveness research could help identify which subsets of patients might benefit more from warfarin versus other anticoagulants, guided by predictive modeling.
- Implementation Science Study:** Finally, studying how these models can be implemented in clinical workflows and their impact on clinical decision-making and patient outcomes could be crucial for their adoption and effectiveness in real-world settings.

These recommendations aim to not only enhance the predictive accuracy and clinical relevance of the models but also to ensure their practical applicability and potential for improving patient care in anticoagulation therapy.

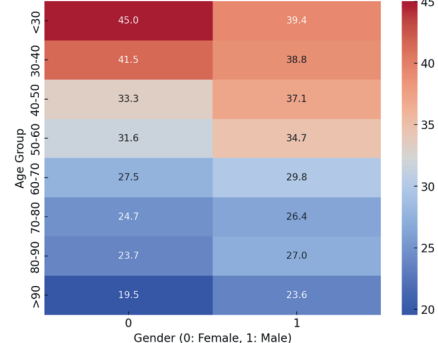
**B**

**You**  
Show me your most creative analysis that can be done with this dataset. Include a visualization about it.

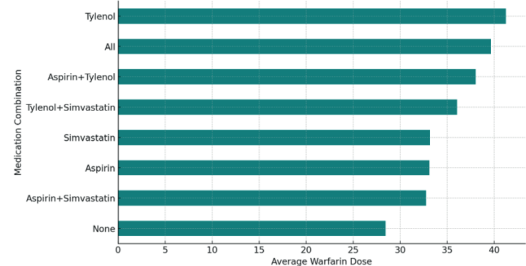
**Data Analyst**  
Finished analyzing



Average Warfarin Dose by Age Group and Gender



Average Warfarin Dose by Medication Combination



**Figure 6.** Additional analysis recommendations and creative analysis examples using Code Interpreter. (A) User request for recommended analyses and ChatGPT's suggestions. (B) Visualizations from the user prompt for the most creative analysis the chatbot can perform.

## DISCUSSION

The findings of this tutorial highlight the significant potential of LLMs in streamlining data analysis workflows for researchers. By using an LLM chatbot with natural language interactions, we were able to achieve similar analyses to those performed in the previous tutorial, demonstrating the power of LLMs in assisting with data analysis tasks. Chatbots can speed up monotonous coding tasks and teach new skills [7], addressing the previous burden of computer programming for life scientists [8]. The experimental result that GPT-4



can perform end-to-end data analysis with given datasets in various domains, similar to experienced data analysts [9], further underscores the potential of LLMs in data analysis.

The democratization of data analysis for non-specialists through tools like CI could lead to a shift in focus for scientists towards higher-level tasks such as designing questions and project management [10,11]. This development has the potential to revolutionize the way researchers approach data analysis, enabling them to focus on the critical aspects of their work while leveraging the power of LLMs to handle the technical details.

However, it is crucial to address the limitations and potential concerns associated with the use of LLMs in data analysis. Researchers must nurture critical thinking to complement LLMs instead of over-relying on them [11]. The development of locally deployable API-based systems for chatbot-aided bioinformatics applications is important to address privacy and security concerns [10]. Additionally, LLMs may make hallucinations, especially with numbers, and may get over-confident assumptions from the data [9]. It is also important to consider the potential inequity if LLM technologies are paywalled or geolocked, emphasizing the importance of making these tools accessible to a wide range of researchers and institutions [11]. Addressing these concerns will be crucial for the responsible and equitable integration of LLMs in data analysis workflows.

In conclusion, this tutorial demonstrates the potential of LLMs to revolutionize data analysis by enabling coding for data analysis to be done in natural language interactions with a chatbot. As LLMs continue to improve and integrate with analytical tools, data science may experience a transformation similar to the shift from manual to automatic transmission in driving. In the near future, we may even witness a level of “full autonomous driving” in data science, where LLMs can handle complex data analysis tasks with minimal human intervention. As we move forward, it is crucial for educators and policymakers to consider the future directions of data science and its education in light of the advancements in LLMs. By proactively addressing the challenges and opportunities presented by these powerful tools, we can ensure that the benefits of LLMs are utilized with proper supervision and responsibility.

## SUPPLEMENTARY MATERIAL

### Supplementary Data 1

Chat log with Data Analyst. The chat log can be also found at <https://chat.openai.com/share/c764ebb0-aca5-4167-9f9a-af8d1634e7d1>

## REFERENCES

1. Barone L, Williams J, Micklos D. Unmet needs for analyzing biological big data: a survey of 704 NSF principal investigators. *PLoS Comput Biol* 2017;13:e1005755. [PUBMED](#) | [CROSSREF](#)
2. Kocoń J, Cichecki I, Kaszyca O, Kochanek M, Szydło D, Baran J, et al. ChatGPT: jack of all trades, master of none. *Inf Fusion* 2023;99:101861. [CROSSREF](#)
3. Nordling L. How ChatGPT is transforming the postdoc experience. *Nature* 2023;622:655-657. [PUBMED](#) | [CROSSREF](#)
4. Zheng T, Zhang G, Shen T, Liu X, Lin BY, Fu J, et al. OpenCodeInterpreter: integrating code generation with execution and refinement. *arXiv*. February 28, 2024. [CROSSREF](#)

5. Ahn S. Building and analyzing machine learning-based warfarin dose prediction models using scikit-learn. *Transl Clin Pharmacol* 2022;30:172-181. [PUBMED](#) | [CROSSREF](#)
6. Klein TE, Altman RB, Eriksson N, Gage BF, Kimmel SE, Lee MT, et al. Estimation of the warfarin dose with clinical and pharmacogenetic data. *N Engl J Med* 2009;360:753-764. [PUBMED](#) | [CROSSREF](#)
7. Merow C, Serra-Diaz JM, Enquist BJ, Wilson AM. AI chatbots can boost scientific coding. *Nat Ecol Evol* 2023;7:960-962. [PUBMED](#) | [CROSSREF](#)
8. Piccolo SR, Denny P, Luxton-Reilly A, Payne SH, Ridge PG. Evaluating a large language model's ability to solve programming exercises from an introductory bioinformatics course. *PLoS Comput Biol* 2023;19:e1011511. [PUBMED](#) | [CROSSREF](#)
9. Cheng L, Li X, Bing L. Is GPT-4 a good data analyst? arXiv. October 23, 2023. [CROSSREF](#)
10. Wang L, Ge X, Liu L, Hu G. Code Interpreter for bioinformatics: are we there yet? *Ann Biomed Eng* 2024;52:754-756. [PUBMED](#) | [CROSSREF](#)
11. Tu X, Zou J, Su WJ, Zhang L. What should data science education do with large language models? arXiv. July 7, 2023. [CROSSREF](#)