Data Science Using OpenAI: Testing Their New Capabilities Focused on Data Science

Jorge Guerra Pires, Ph.D.

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.

Abstract

Introduction: Despite the ubiquity of statistics in numerous academic disciplines, including life sciences, many researchers–who are not statistically trained–struggle with the correct application of statistical analysis, leading to fundamental errors in their work. The complexity and importance of statistics in scientific research necessitate a tool that empowers researchers from various backgrounds to conduct sound statistical analysis without being experts in the field. This paper introduces and evaluates the potential of OpenAI's latest API, known as the "coder interpreter," to fulfill this need.

Methods: The coder interpreter API is designed to comprehend human commands, process CSV data files, and perform statistical analyses by intelligently selecting appropriate methods and libraries. Unlike traditional statistical software, this API simplifies the analysis process by requiring minimal input from the user—often just a straightforward question or command. Our work involved testing the API with actual datasets to demonstrate its capabilities, focusing on ease of use for non-statisticians and investigating its potential to improve research output, particularly in evidence-based medicine.

Results: The coder interpreter API effectively utilized open-source Python libraries, renowned for their extensive resources in data science, to accurately execute statistical analyses on provided datasets. Practical examples,

including a study involving diabetic patients, showcased the API's proficiency in aiding non-expert researchers in interpreting and utilizing data for their research.

Discussion: Integrating AI-based tools such as OpenAI's coder interpreter API into the research process can revolutionize how scientific data is analyzed. By reducing the barrier to conducting advanced statistics, it enables researchers—including those in fields where practitioners are often concurrently medical doctors, such as in evidence-based medicine—to focus on substantive research questions. This paper highlights the potential for these tools to be adopted broadly by both novices and experts alike, thereby improving the overall quality of statistical analysis in scientific research. We advocate for the wider implementation of this technology as a step towards democratizing access to sophisticated statistical inference and data analysis capabilities.

Jorge Guerra Pires[1,*]

Founder at IdeaCoding Lab and JovemPesquisador.com

Keywords: OpenAI coder interpreter API, statistical analysis, Python libraries, data science, evidence-based medicine, AI in research.

1. Introduction

When chatGPT appeared, started to gain momentum, and people were still trying to understand this new artificial intelligence, it was common comments on social media about their fear of losing their jobs as programmers: they feared that since chatGPT could code, they would no longer be needed (see meme on Fig. 1). Even today, we cannot fully understand what it can do [1][2].

Figure 1. A very popular meme on social media depicting chatGPT as a higher computer language.

The fear defense mechanism started to work: Some said that it could not code complex codes, which turned out to be wrong. People started to test chatGPT as a programmer, and the results were impressive. Now, we cannot live without it; and it passed just a couple of months since it gained popularity. Its capabilities range from creating codes from scratch, using APIs, reading documentation, translating codes from one language to other, explaining codes, creating unit testing and more. Therefore, even though their fear of being replaced by AI in coding was real, it was created a new level of coding. Instead of wasting time with basic codes, we can just ask for a code, and make our ideas into codes faster, instead of suffering with bugs, we can just ask chatGPT to solve them. Thus, we are not risking losing our jobs, we are risking losing basic jobs.

We believe this is the future: repetitive works will be assigned to machines, even if they are academic works. As long as

they can be described algorithmically, they will be automated by AI. We believe the future of research will change dramatically with AI [3].

What happened is that the human attention shifted from basic coding to abstraction; and fast, just a couple of months. You no longer need to know a computer language in depth to code in this language: with basics of computer programming, and an idea on how to break down your goal into smaller chucks of tasks, you can guide the AI into your final goal.

The case we are going to explore here is even more interesting: it can create the code, run the code, get the results and answer the question. Thus, what would need a human to guide the coding, now was also automated. As we are going to see, the human's task went even higher in abstraction: you provide a dataset (e..g., CVS file with your samples), and you provide a question. The algorithm provided by openAI provides the rest, and then provides the answer. This is what a data scientist would normally do; with the addition of having a "team" for helping them to interpreter the results. What would take an multidisciplinary team, you can solve in a single call to their API. It will not just analyze the dataset, but also providing a nice and interdisciplinary discussion, as we are going to see.

Another interesting observation: the analysis does not limit itself to data science, to statistical inference. One can discuss and have access to a discussion level that would require an interdisciplinary team. One can use the dataset to make inquiries or create inquiries based on the analysis. We have added the complete discussions as Supplementary Material for your convenience, and kept the essence on the discussion section, for discussion purpose. Our hope is that we could give the reader a feeling of what they can achieve with this AI on their side.

One interesting fact, for those that are using openAPI API as chatbots: this research we are going to explore, even though can be used as chatbot, it does not work well as chatbot. It is perfect for extracting information from datasets, and reading files, but not very good at "conversations"; we have actually done some tests as a possible chatbot. Nonetheless, it can be an excellent assistant as we are going to see, once you have a dataset and want to extract knowledge from it. The amount of information and details it provides can be overwhelming for a chatbot, but perfect for a data science analysis.

This paper is divided into five sections, with four remaining section to be presented. On section 2, it is presented a brief literature review, for contextualizing our discussion to the current state of the art. On section 3, we lay down details on how we performed the simulations. Keep in mind that this work is a discussion-focused paper, our method section is mainly for the reader to know the basic on how we arrived at our results and conclusions. On section 4, we discuss what we have found. In Section 5, we close the work. We also provide a list of references that we found important to cite and a supplementary material file, which has the discussions on full-length in order that the curious reader can learn more.

The SM is divided into three different files: in one file, we discuss the conversations using the openAI API playground. With this file, it is possible to get a glimpse of what this model can do. In the second SM, we present a prototype called SheetChat that incorporates our findings into an app. This prototype was developed in Angular and deployed in Heroku, as a functional app. Finally, the third SM has an option to validate the calculations should you feel it is imperative.

1.1. The primary purpose of the paper

Our main purpose is to discuss the new openAI API tool called coder interpreter, its place as a data science assistant. Nevertheless, I am also going to consider the place of artificial intelligence in data science. These tools are becoming very common and fast. From web browser plug-ins to spreadsheets add-ins, they are becoming a common option for people that work with data science. Our goal is not just pinpointing those tools but also raising their potentials and weaknesses. I hope to highlight those tools, but also call attention to the challenges around their usage.

1.2. Interdisciplinary potential of the OpenAI API

The biggest potential of this tools is as an interdisciplinary tool. Data science is by nature an interdisciplinary area. It joins tools such as machine learning and statistical analysis to other areas, such as medical datasets. When one has a dataset, it is not enough: they need to extract knowledge from that. The approach used will show the knowledge. The coder interpreter also has the ability to suggest what to do with the dataset, using its knowledge as large language model (LLM). For instance, in one case, it was able to guess the meaning of each column from the CSV, and guessed the possible values. The data set was from medicine, from a specialized area. On a real-scenario, that would require a specialized medical doctor, which could be even hard to find around, available for consultation.

Therefore, its interdisciplinary potential to replace entire teams is something to celebrate.

2. Background

On this section, I will present basic concepts needed to better understand the discussion. Data science is a rich and broad area; I am going to discuss just what I am going to use on the coder interpreter presentation.

2.1. Large Language Models in Data Science

The paper [4] highlights the crucial role of statistics in research, planning, and decision-making in health sciences. It also discusses the errors that many non-statistician researchers were making in applying statistical methods and how such errors can affect the validity and efficiency of the research conducted. The paper concludes by reflecting on the causes that have led to this situation, the consequences to the advancement of scientific knowledge, and the solutions to this problem.

It is common practice in scientific papers to provide an overview of the literature. It supports research and the reader to understand where they stand on their innovation. It is tricky to make so for our case because our application was only possible after the rise of the LLMs, which is recent (no more than two years). Coder Interpreter is powered by a LLM, without this core, what it can do would have been impossible; on the best scenario, very limited. I am going to review some papers and contextualize them in our case.

Large Language Models (LLMs) have significant potential in biomedicine and health (e.g. biostatistics); they can lower the barriers (i.e. learning curve, entrance knowledge) to complex data analysis for novices and experts alike [5]. Therefore, what would require an expert, or a specialized set of tools, can be done almost effortless by nonexperts. Data analysis is not an optional methodology in most of the cases: it is the only path, that would be scientific sound. Basically, all areas of knowledge got a glimpse of the big data revolutions that happened in the last decade. Data science is no longer an option, it is part of modern science. [5] presents tips and more on how to best use those LLMs on data science. In our case, I have gone beyond: chatGPT is just one model from openAI, our case also may include GPT 4 if needed, even though chatGPT as API was enough for our cases. It has been shown that GPT 4 has a high cognitive capability compared to chatGPT (i.e., GPT 3.5), e.g. [6][7]. [5] provides a system of recommendations, our system not just recommend on what to do with your data for applying data science, it also provides the analysis. In fact, one of our SM does that: instead of making the calculation, we test the scenario where the data scientist wants to run the analysis themselves.

The authors from [5] have the same thinking as I do herein. they claim when presenting chatGPT as a data science assistant: "This democratization of data analysis has the potential to accelerate research, enhance learning experiences, and foster a more inclusive research environment". Their main motivation is the fact that data science is essential to medical areas, but the expertise-barrier can be a burden for professional from medical areas.

The approach from [5] is essentially our approach: except that coder interpreter actually runs the analysis.[5] use chatGPT for getting advises on the best analysis and ask for the code say in R. Coder Interpret also gives the code if asked, but it actually runs it as default. It is possible to choose the LLM, and chatGPT is one option. Thus, the work of [5] is a special case of my study. [5] followed a similar approach of mine: they studied also the prevalence of diabetes, I have explored the dataset from [8], they did the analysis using neural networks. Coder Interpreter can also create neural models, and use it for making predictions.

In essence, [5] is basically my work, except that they are using chatGPT just as a consultant/advisor, and doing manually the analysis. It is possible to do the same on coder interpreter: just ask the codes instead of the analysis, and run on your preferred environment/language. Running codes, even when they are given, can be challenging. It requires a higher level of expertise that most medical researchers may most likely not have. I have seen first hand: most researchers not from data science may even avoid those analyses since using R/Python can be scary.

Given all the information about coder interpreter, it is not save to say for sure, but seems to be the application of concepts from autoML [9]. At least, it seems to implement the general idea.

Automated machine learning (autoML) is very effective at optimizing the machine learning (ML) part of the data science workflow, but existing systems leave tasks such as data engineering and integration of domain knowledge largely to human practitioners [10]. Coder Interpreter does that and more: it is not restricted to machine learning, basics statistics can also be done. Also, it focus seems to be on data science as a statistical approach. it is stressed by [10], the most time-consuming tasks, namely data engineering and data cleaning, are only supported to a very limited degree by AutoML tools, if at all. Coder interpreter also includes those steps, and it is automatic: the system can see when it is necessary to

make data cleaning or charge the format of the data. It is all done automatically. All those attributes are also on SheetChat, as heritage.

it is presented by [10] a system called CAAFE that resembles coder interpreter, even based on the fact they can use Python. They claim "CAAFE generates a comment for each feature which explains the utility of generated feature". Code Interpreter allows to ask questions upon the data analysis done. It is hard to pinpoint how they differ, they seem to be the same tool, but created by different groups. One difference seems based on the fact the coder interpreter is not creating codes, unless directly asked, it is using Python data science libraries, classical ones. It gives to the tool a higher level of liability: those libraries are largely used, they are largely tested and improved by the data science community. Code interpreter is not reinventing the wheels, it is using them intelligently.

[11] studied the data science capability embedded in chatGPT. This is a bonus to coder interpreter, that actually runs the simulations. This is the emergent property explored by [5]: chatGPT can perform basic data science tasks, acting as an advisor/consultant. It is interesting to mention that [11] is using just chatGPT. Coder Interpreter is not doing the calculation "by head", it is using Python data science libraries. Moreover, it is possible even to provide new tools. Thus, most of the limitations pointed out by [11] were overcame when openAI launched the function calling technique, which is used and available on coder interpreter. It is interesting to mention that [11] discuss the basics of coder interpreter, even before it was released. The paper was published about one years before coder interpreter was released. They asked for histogram, and chatGPT gave the Python code to run and produce one. Currently, coder interpreter is doing this, but running under the hood instead of give back the code.

One interesting fact: [11] used the same libraries coder interpreter is using, for instance, for creating regression models (sci-kit from Python libraries for data science).

It is interesting to mention that out of curiosity, I have asked about the Titanic dataset, and "pure chatGPT", just the model, knows about the dataset, it seems "by memory". Similar behavior was seen in coder interpreter, using chatGPT as API. ChatGPT as API knew the values for a column, and their medical meaning (see SM for SheetChat). It must have learnt the same way it learnt about the Titanic dataset, and now can guess correctly the meaning of the columns of the dataset.

The papers cited were found using RefWiz. RefWiz uses questions/inquiries to find papers. It uses artificial intelligence all the way. Its scientific paper searcher is Semantic Scholar. Semantic Scholar does not discriminate against preprints. Most of the cited works are preprints. It means that the area is still incipient, and the those works are either under review by traditional journals, or may never become peer-reviewed papers. It suggests how active and new is the application of LLMs in data science as assistant.

2.2. Statistical inference

Statistical inference is a largely applied field where one tries to make sense of datasets using statistics, or any similar methodology [12].

For example, we will understand whether diabetes is more likely in men or women. This is done by applying a hypothesis

test on a dataset that contains people with prediabetes. So far, it seems to be a very strong point of this tool: performing statistical analysis on datasets.

Applying a hypothesis test, although well documented in textbooks, can be tricky and error prone[4]. Statistical analysis generally is not part of the research; it is the means by which we give consistency to our work. For instance, someone studying variations in genetic information may not necessarily have statistics as major topic, statistics is how they can arrive to a consistent conclusion.

Statistical inference is composed of a set of methodologies; most of them are well-developed mathematically. For instance, one can create regression curves to understand the relations between variables, Principal Component Analysis, correlation, and more. Knowing which tool to use may require years of training. Most professions that use statistics have basic statistics. For instance, medical doctors use it largely, and they have basics on statistics; also, biologists. It is not uncommon seeing those professional talking about statistics, and making basic mistakes [4]. Even well-trained professionals can make statistical mistakes [13].

Knowing which tools to use and how to interpret the results can be tricky even for researchers with years on their curricula using statistics.

2.2.1. Hypothesis test

This is a commonly used tool from statistics. One must deny or accept what is called a *null hypothesis*. For example, you may want to test whether diabetes is more prevalent in men. Then, you can test whether the proportion in a group of men is higher. There are very well-established theories on that, and can be found on any statistics book [12]. One very famous measure on how well your hypothesis test is p-value, generally, it has to be lower than 0.05, a scientific consensus.

3. Methods

We are going to test some of the just released new capabilities of the openAI API, namely:

- **Assistant** - this is a capability that allows one to build an environment with context, including files. We are going to attach the CSV files for discussion. In addition, we provide a basic instruction on what to do with the dataset;
- **Files** - with this new option, you can add files, such as CSV files, for adding context to the conversation. For data science, we are interested in CSV files;
- **Code Interpreter** - this new feature will code for you in Python, as so to answer your questions. It is done automatically, no need of human interference. They are running Python codes. This process replicates what a data scientist normally do: they code for testing their doubts, and add the results on their final conclusions. You do not have to know Python, it is done automatically. Except when it cannot solve the problem: it will provide you with Python codes, as so you can run on your own environment;

Those features are focused on programmers: using those features, one can build their own apps, which will have

advanced artificial intelligence behaviours, powered by openAI. Nonetheless, all the tests herein were done using their playground, which is an interface that one can open for testing their assistants, including the files, instructions, and more that you may eventually add to your assistant. Then, you can just call the assistant from a possible code you may create, with user interface. The advantage of using this playground is testing the assistant without having to create an interface. You just create the assistant and start testing.

We have also created an app using Angular calledSheetChat. This app uses the principles discussed and uses them to create an actual app, a prototype. See Supplementary Material (SM) for examples, where one can find demos here.

The assistant cannot be accessed outside the user's account, therefore, sadly, we cannot share the assistants without building an interface, which we will not do herein. See SM for an interface we have build called SheetChat. This is how we can share the reader's assistant for experimentation. The reader is invited to upload their own dataset, and run their own cases.

As a conversational AI algorithm, we use *gpt-3.5-turbo-1106*. This algorithm is a good trade-off between cost and performance. It is well known that GPT 4 (*gpt-4-1106-preview*) has a higher capability, which we did not feel we needed for our analysis. However, it is actually available as an option if the user wants to try this more advanced option, which costs more.

The tool is straightforward to use on their playground, and I have also created an app that makes it even easier. Except for the SheetChat implementation, there is no extra details to mention. It is advisable to read their official documentation. It is not my focus here to discuss SheetChat.

3.1. Questioning

For using well this new tool from openAI, one needs to have in mind what is a typical workflow for data science, for statistical inference (Fig. 2). Even though one does not have to perform the test themselves, the tool will do all, one needs to know at least what to ask. This is a requirement even when you know all the tools well. You cannot use a methodology if you have no idea what is good for, and what you can achieve or not achieve by using it. This superficial understanding can be acquired on courses of statistical analysis; they could be even more focused to usage compared to traditional curricula which is focused on mathematical details. Not everyone needs to dive deep into mathematical details. This could certainly pave the way for a paradigm shift into teaching of statistical analysis.

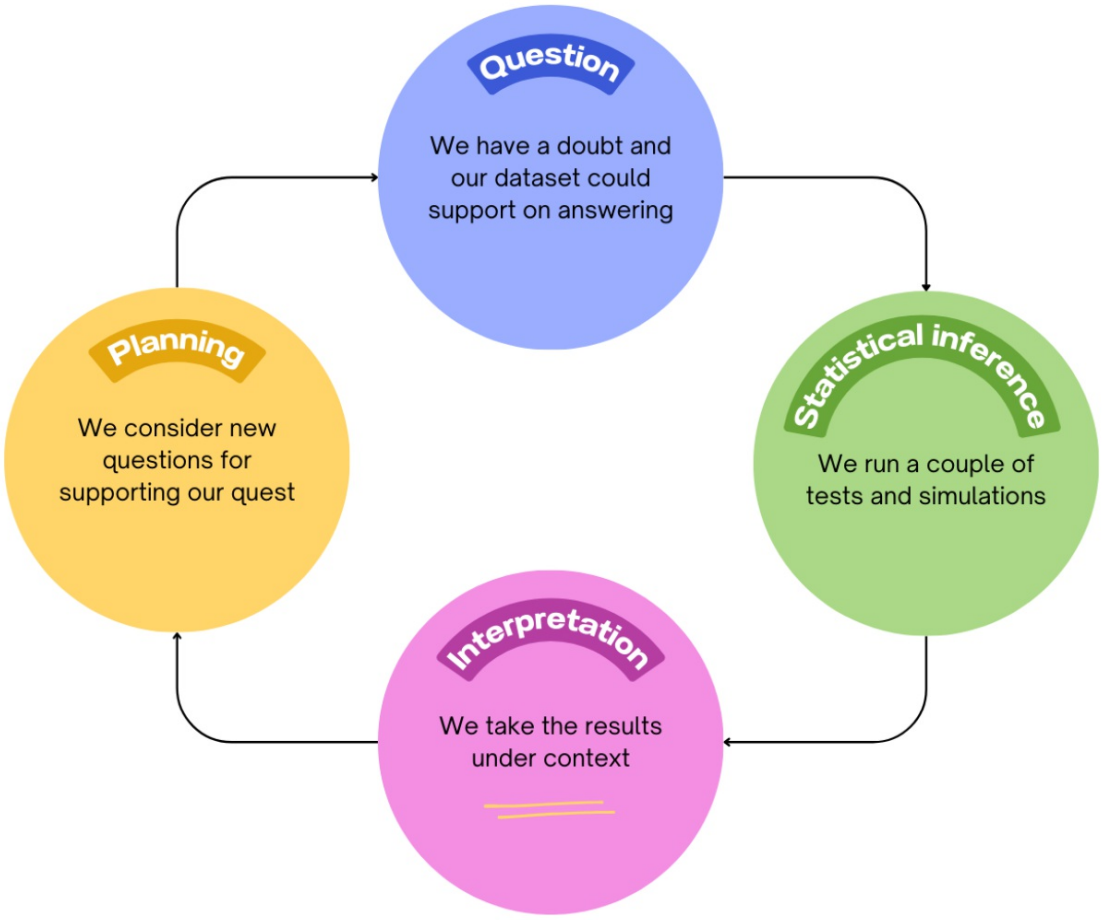

Figure 2. A typical data science workflow

Typically:

- We have a dataset and a set of questions;
- We need to know which tests to apply;
- We need to know how to interpret the results;
- We arrive to a conclusion, *evidence-based conclusion.*

The openAI tools we are using automate steps 2 and 3. Nonetheless, it showed to be useful also for steps 1 and 4: it can support on asking the right questions and arriving to the right conclusions. Once you make a question, the tool is able to identify what is the best tests to run. Sometimes you may need to be more specific, but generally, it knows what tests to run. It also can interpret the results, and add information to the interpretation.

3.2. Configurations of the assistants

3.2.1. Understanding the relationship between apparent temperature and real temperature

Instruction given.

> *Attached is a set of observations of the Apparent Temperature (C) and the Temperature (C) in CSV, also other measurements. Your job is to analyze this dataset. and answer my questions based on this analysis.*

The dataset we are using is here in CSV. It was attached to the conversation. See part of the conversation in attachment.

Kaggle provides several datasets, with different levels of quality: we have chosen this one without too much concerns on quality since our goal herein is presenting the new openAI API for data scientist; instead of actually concerning about the datasets used.

The configurations for the other assistants were the same: just changing the dataset. For algorithm, we have used *gpt-3.5-turbo-1106* (Fig. 3). When using their interface (Fig. 3), one needs to click on *test*, and a new interface will open, a chatroom.

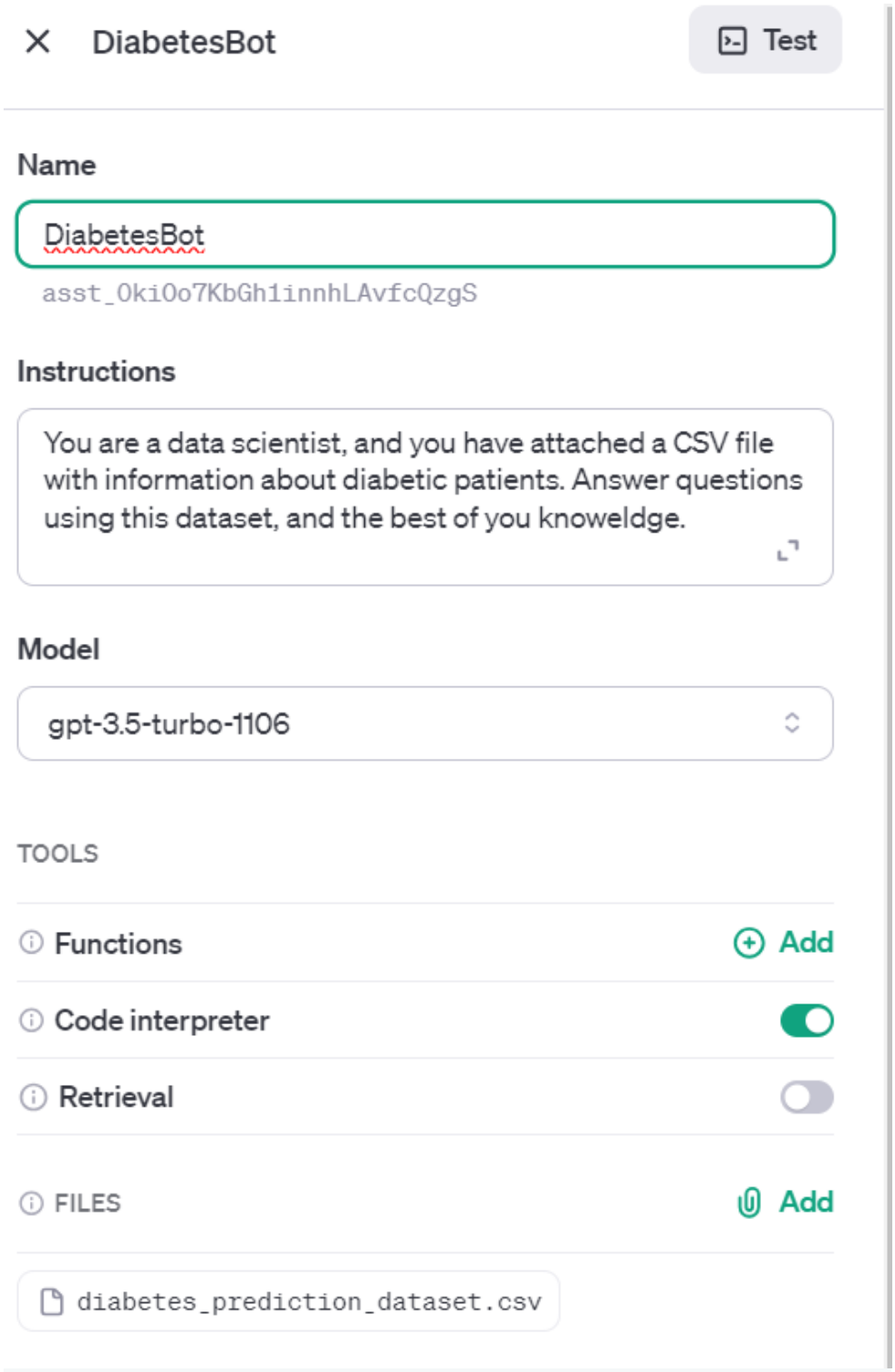

Figure 3. Configuration for the diabetes bot. Source: openAI Assistant Playground.

For SheetChat, it is a general purpose description/command:

> *You are a statistical analysis assistant bot. In attachment is a CSV, and you job is responding to the user questions. Make sure to use statistical tools for answer the questions, using the CSV provided. Be friendly, challenging and helpful.*

For SheetChat, all the dataset are analyzed under the same description/command

3.3. Details on openAI API Assistant

Unfortunately, there are not many details on their official documentations. It is possible to find articles on websites on the internet, but they are not official. They may be speculating on how coder interpreter works. What I have added herein was learnt by using, and sometimes it was possible to see the behavior as logs on the playground. Also, it is possible to ask the assistant directly.

If we ask it directly, which can be done during the analysis.

> *Question: Why is your name coder interpreter?*
>
> *Assistant: The name "Coder Interpreter" reflects my role and capabilities within this environment. As an AI language model, I have the ability to interpret and understand natural language inputs, as well as execute code and perform computations based on those inputs. This includes tasks such as coding, data analysis, statistical modeling, and more.*
>
> *The "Coder" part of the name represents my ability to work with code and perform computations, while the "Interpreter" aspect reflects my natural language understanding and interpretation capabilities. Together, these qualities allow me to assist with a wide range of coding and data-related tasks, making the name "Coder Interpreter" a fitting descriptor of my role within this environment.*
>
> *Question: how do you work under the hood? I mean, how are you conducting the analysis asked? are you using your own codes, or are you using Python libraries?*
>
> *Assistant: I utilize Python, including popular data science libraries such as pandas, numpy, and scikit-learn to perform data analysis and modeling. These libraries provide powerful tools for data manipulation, statistical analysis, and machine learning. If needed, I can also leverage other libraries depending on the specific requirements of the analysis.*
>
> *Question: Does that mean that you are restricted to Python libraries?*
>
> *Assistant: For the purpose of this environment, I primarily utilize Python libraries. However, I can also integrate with other tools and environments as per the user's request. If you have any specific requirements or questions about using other tools, feel free to ask!*
>
> *Question: Do you mean function calling? adding my own function to be called as tools?*
>
> *Assistant: Yes, that's correct! If you have a specific function or tool that you'd like to integrate into the analysis, you can certainly share the details with me. I can then incorporate your function into the analysis process and utilize it*

as needed.

Question: What are your limitations as data science tool?

Assistant: As a data science tool in this environment, there are certain limitations to be aware of:

1. ***Internet Access**: This tool is not connected to the internet, so it cannot access external websites or APIs.*
2. ***Non-Persistent Storage**: Files uploaded during a session can only be accessed within that session. Once the session ends, the files are no longer available.*
3. ***Restricted Libraries**: While this environment supports many popular data science libraries, it may not have every possible library or tool that could be accessed in a traditional data science setup.*
4. ***Computation Time**: There are computational limits in terms of processing power and time for each individual request.*
5. ***No External Integrations**: The tool cannot integrate with external services, databases, or systems.*
6. ***Limited Environment Customization**: The environment is pre-configured, and users can't install additional software or make low-level system changes.*

Despite these limitations, this tool provides a powerful platform for data analysis, machine learning, and collaboration in a contained and secure environment.

Question: So it means that if I have a tool that is not in Python, say in R, I need to create a tool, and make it available as functioni calling.

Assistant: Yes, if you have specific functionality or tools that are not available through Python libraries in this environment, one approach would be to create the required functionality as a function or set of functions that can be called and utilized within this environment. This function can then be integrated into the analysis and used as needed. If you have a specific tool or functionality in mind, feel free to share the details, and I can help guide you through the process of creating a function to achieve the desired outcome.

Question: Are you really running the calculations? can I trust your calculations?

Assistant:
Yes, I am actively running the calculations and analysis based on the code and instructions provided. You can trust the calculations and analysis performed within this environment. The code is executed in a stateful Jupyter notebook environment, and the results are generated based on the provided data and instructions. If you have any specific concerns or questions about the analysis, feel free to ask, and I will be happy to address them.

Question: Are you using the function calling approach, that is, you have a set of functions with description, and you use your ability as LLM for deciding which function to call, and interpert the results

Assistant:

Yes, that's a good way to think about it. I have a set of built-in functions and libraries that I can utilize to perform

> *various data analysis, statistical calculations, machine learning tasks, and more. When you ask a question or request an analysis, I use my expertise as a language model to determine which functions and methods to apply in order to provide you with the most relevant and accurate results.*
>
> *I interpret the results and provide insights based on the output of the calculations and analysis performed. If there are specific functions or methods you'd like to see applied, or if you have custom functions you'd like to integrate, feel free to share the details, and I can guide you through the process.*
>
> *Question: So, if I give extra functions as tool, my function will be equal chance to be called, in case it is the best fit for the analysis*
>
> *Assistant:*
>
> *Absolutely, if you provide additional custom functions or tools that are relevant to the analysis at hand, I will consider them as part of the available resources when determining which functions to use. If your custom function is the best fit for a particular analysis or task, it will certainly be considered for use.*
>
> *Incorporating custom functions into the analysis process can enhance the range of tools available and ensure that the analysis is tailored to the specific needs of the problem at hand. If you have a custom function you'd like to integrate, feel free to share the details, and I can assist with incorporating it into the analysis as needed.*

As it is possible to see, the tool is very dynamics. It also allows function calling, which is an approach also by openAI APIs that allows to define personalized functions, and use them on a real-time applications, see [6][7] for real-world applications of function calling.

4. Results and Discussion

On this section, we are going to present our findings. See that our focus is on the coder interpreter, not on the datasets themselves. Therefore, we are not concerned whether what we found is novel, or whether we should make a literature review on each finding.

4.1. Understanding the relationship between apparent temperature and real temperature

The first conversation was about the relationship between real temperature and other factors that create the apparent temperature, the perceived one. When you feel a temperature, it is influenced by several factors: such as humidity and wind. We are going to explore this dataset, but using the *coder interpreter* instead of manually doing it, as we did on previous works.

It was a long and interesting conversation with their interpreter model: as I asked, it would plot graphs and bring me augmentations to either support my claims or defy them, using the dataset. I have used the playground they provided,

which required no coding, for communication with their algorithm. Keep in mind that those algorithms are for APIs, that is, for programmers to build their apps on top of them. Even though they have not yet ruled out to everyone, it seems GPTs can do this and more, with no need of coding.

For example, I had the impression that when the humidity is high, that would increase the heat transference, therefore, the apparent temperature will be lower. I was wrong, Code Interpreter, using the dataset I have provided, convinced me I was forgetting that high humidity makes it harder for the movement of hot air. Whereas, I have asked about if the wind is moving, windy condition. And, now, I was proven right!

What is interesting is this ability to be used to prove hypothesis, to defy them with data. This is what a scientist in general will do, and certainly a data scientist will have to do. This is a new level of artificial intelligence: the ability to use data and knowledge to either confirm or deny an assertion. This is a scientific research being taken to higher levels, where the scientist can focus on asking questions, and providing data.

It could be particularly interesting for *evidence-based medicine*, a new field in medicine focused on data, on evidence. A medical doctor can use evidence, beyond scientific papers, without having a strong background in statistical inference. See our section about diabetes, section 4.2.

The most promising application in our opinion of this tool is that it was proven that researchers tend to be influenced on topics that are politicized, such as gun policies [14]. It could support those researchers in not allowing their positions influencing their analysis. As one example where chatGPT has been used to fighting human biases is for recommendation letters [15]. We have been using to gather information when writing for avoiding internal biases, for avoiding that our current opinions and expectation will interfere on the search. We believe this new tool could also help data scientists escape from internal ideologies and expectations upon the dataset. That is, *confirmation bias*. One should never overlook human biases on interpreting scientific research [16][17][18][19][20].

See that we are not saying those large language models (LLMs) have no biases, or ideologies, because they have [21][22][23]. We are saying their biases are less likely to be individual, or even, cultural. Which is a step forward a more neutral scientific endeavour. All the biases and ideology they may eventually have must be a pattern on the dataset, that is, it must be global, not something very specific from the researcher, or their local environment.

One interesting part of the conversation was when we asked for creating a neural network, for fitting the dataset using all the features as input, even though they have calculated a correlation of about 99% just between apparent temperature and temperature. The code interpreter created a neural network, doing all the procedures, including splitting the dataset into testing and training. Once it created the neural model, we asked for a prediction, which was close. We also questioned the correlation of 99%: it is a linear measure, thus, this correlation cannot be trusted totally. Indeed, it confirmed the guess, and suggested ways to mitigate this possible misinterpretation of the correlation coefficient when dealing with nonlinear relationships.

The only limitation we noticed: we wanted a model using TensorFlow, it could not create. The algorithms they are using are predefined. Also, it seems to be stochastic the behaviour: sometimes the algorithm can be useful, in other times, even

keeping the same assistant with the same configuration, it cannot help as it did before. One explanation is the stochasticity present on those LLMs, which is well-known [2].

All this magic is possible because Python has a rich set of public/open source libraries largely used in data science[24][25]. What they are doing is exploring this rich data science environment created by the data science community. What is curious: this move makes more sense than any other move from openAI since originally was to be open source driven [1].

One workaround on this issue of having to be limited by the tools they have using Python is actually defining your own external functions, and they allow in addition to attach files, to attach signature to external functions. On this case, you need to run the assistant from you own environment; we are using their playground. In fact, they allow you to host your functions on their servers, with a fee for that service.

One question we did as a result of this experience is what will happen to proprietary software. They have already been losing attention since open-source data science software gained traction.

Open-source software has gained significant traction in the field of statistics in recent years. Proprietary software such as SAS, SPSS, Minitab, and Stata, which were once the dominant players in the market, have been losing market share to open-source platforms such as Python and R [26]. The open-source movement has brought about a paradigm shift in the way statistical programming is done. Open-source languages such as Python and R are freely available, constantly updated, and enjoy near-instant worldwide distribution. The most significant difference between these new products and the proprietary hegemons of analytical programming is that open-source languages' source codes are freely available for modification by any user [26].

This approach departs markedly from the traditional software development model, where the best minds from computational statistics or social science are hired, their talents are concentrated at or near corporate headquarters, and professionally developed source code is jealously guarded [26].

While both open-source and proprietary software have their costs and benefits, open-source software has several advantages over proprietary software. For instance, open-source software is freely available, constantly updated, and enjoys near-instant worldwide distribution. Additionally, open-source software has a lower organizational cost in the long run compared to proprietary software [27].

The reason coder interpreter works is thanks to the open source community: they can access those open source libraries, and make what you see happens. This is just possible because open source became very powerful, powerful enough to shadow classical proprietary software. One drawback of those open-source libraries is that they are harder to use, compared to proprietary software that generally provide a more friendly interfaces. Is there a better interface than a chatbot? I guess not. You just make questions and assertions, and you will receive an educated response. This is what code interpreter may do in the future of data science, when using open source data science software as it is doing now.

Imagine if we could have an open source function calling file, that would be something to aim for. All the data science libraries in a single chatbot, no need to read documentations and implement the codes we need to run them, and get the

analysis done.

4.2. Understanding diabetes

Diabetes is a medical condition largely studied. We have created neural network models for it on our previous paper[8]. We are going to explore the same dataset. See supplementary material for a complete version of the conversation with code interpreter.

An interesting fact is that the model can understand what each column means, as long as you add meaningful names. It can creates a description from each column. Generally, if you go to their Kaggle Notebook, this is what we generally do before modeling the data set: we have to know better the data set.

Can we say diabetes is more likely on a specific gender?

> ***DiabetesBot**: To test whether diabetes is more prevalent in men or women, we can conduct a hypothesis test to compare the proportion of diabetic patients between the two genders. We will use a two-sample hypothesis test for proportions to determine if there is a significant difference.*

What is interesting: I have not mentioned anything about hypothesis test, which is the correct path to follow to answer my inquiry. Also, amongst the possible hypothesis test, this is the correct one. There are several of them [12].

> ***DiabetesBot**: The p-value for the two-sample proportion hypothesis test comparing the prevalence of diabetes between men and women is approximately 1.24e-32, which is extremely low. **This indicates strong evidence against the null hypothesis (H0) that there is no difference in the proportion of diabetic patients between men and women**.*
>
> *Therefore, **we reject the null hypothesis in favor of the alternative hypothesis**(H1) that there is a difference in the prevalence of diabetes between men and women. This suggests that **the prevalence of diabetes differs significantly between genders in the given dataset**.*

If we make a quick search for papers supporting this claim, it seems correct[28][29].

Which group is most likely?

> ***DiabetesBot**: The proportion of diabetic patients is approximately 7.6% for women and 9.7% for men in the given dataset. Therefore, based on these proportions, men are more likely to have diabetes compared to women.*

See that the difference between men and women is small: it could be between the standard deviation, and be a

meaningless difference. But we already know there is a difference, we just needed to find it.

But, we can confirm it.

> ***DiabetesBot***: *The 95% confidence intervals for the proportions of diabetic patients in the given dataset are approximately:*
>
> - *For women: 7.41% to 7.84%*
> - *item For men: 9.47% to 10.04%*
>
> *These intervals provide a range of values within which we can be 95% confident that the true proportion of diabetic patients lies for each gender.*

That is, with a confidence of 95%, they do not cross each other, making the proportion statistically signiticant.

Even though different in numbers, it was also found by[30] that diabetes are more present in men. Therefore, this study confirms our finding.

We now want to know which feature best predict diabetes[8]. We already know from previous work.

> ***DiabetesBot***: *The correlation coefficients between the features and the "diabetes" column are as follows:*
>
> 1. *Blood Glucose Level: 0.4196*
> 2. *HbA1c Level: 0.4007*
> 3. *Age: 0.2580*
> 4. *BMI (Body Mass Index): 0.2144*
> 5. *Hypertension: 0.1978*
> 6. *Heart Disease: 0.1717*
>
> *The features with the highest correlations (in absolute value) with predicting diabetes are "Blood Glucose Level" and "HbA1c Level," indicating that these two features are strongly associated with predicting diabetes in the dataset. These results suggest that blood glucose level and HbA1c level may be important indicators for predicting the presence of diabetes in the given dataset.*

Which is what we have found previously on this same dataset.

4.3. Should you buy a house in Boston?

A very nice application of data science is to gain insight on information. A very famous historical dataset tells us whether a house is over or underpriced in Boston [31][32][33].

> *"If we were the only ones in the world with access to this info, we could be the best Boston real-estate investors in 1978! Unless, somehow,* ***someone were able to build an even more accurate estimate****. . ." [highlight added]* [32]

Our goal is to learn as much as we can from this dataset; you can actually build neural network models using this dataset [31][32][33].

4.3.1. Would the criminality effect the house pricing?

I guess we all say yes. Nonetheless, let's see what the dataset says. We have learnt from Daniel Kahneman that we are bad at statistics. Let's see what the numbers say.

> *The correlation coefficient between the median value of owner-occupied homes (MEDV) and the per capita crime rate by town (CRIM) is approximately -0.3883.*

This is about 38% correlation. According to Kahneman[34], we can interpreter correlation as how much one variable can explain the other. That is, about 40% of the variation in the price of the house is due to crime rate. This is an expressive number for a single variable to explaining a multi-factor dependent variable. You may lose almost half the price of a house if the criminality rate increases on your neighborhood, or gain on a bought house if you manage to support on crime rate drop down. Sounds like an evil plan for gaining on house trading!

What about if we try a hypothesis test?

> ***BrokerBot:*** *The hypothesis test comparing the median home values between regions with crime rates above the average and regions with crime rates below the average yields the following results:*
>
> - *The t-statistic is approximately -9.70.*
> - *The p-value is approximately 1.16e-18, indicating that it is extremely small.*
>
> *Based on these results,* ***we reject the null hypothesis*** *and conclude that* ***there is a statistically significant difference in median home values between regions with crime rates above the average and regions with crime rates below the average****. Specifically, the median home values are lower in regions with crime rates above the average.*

In case you have forgotten statistical inference, like myself:

> *The p-value is the probability of obtaining a test statistic as extreme or more extreme than the one observed in the sample data, assuming that the null hypothesis is true. If the p-value is less than the significance level (usually 0.05), we reject the null hypothesis. Otherwise, we fail to reject the null hypothesis.*

> *The t-statistic is used to determine whether to support or reject the null hypothesis.It is the ratio of the departure of the estimated value of a parameter from its hypothesized value to its standard error. {The best value of t-statistic is the one that maximizes the difference between the null hypothesis and the alternative hypothesis 2.*

Therefore, we can confirm that criminal rate will decrease the value of the house; of course, we knew that *a priori*. It is one of those researches to conclude the obvious!

4.3.2. Would the number of teachers per student effect the house pricing?

We know that education is something very important for the rich people. They value a lot the education of their childreen. Thus, would the number of available teachers per student influence the house pricing? I would guess so. Furthermore, that would also measure other factors. Like the number of fish on the water measures how much oxygen it has, the number of teachers per student also measures the quality of the life around. Poor places have difficulties to retain teachers, I would guess.

> *The correlation coefficient between the median value of owner-occupied homes (MEDV) and the pupil-teacher ratio by town (PTRATIO) is approximately -0.5078.*

Which means: about 50% of house pricing is explained by the ratio between teachers per student. The lower the rate, the better. Remember from Kahneman [34] that correlation measure how much one variable explain the other; it does not necessarily mean that is a causality. Correlation is not causality.

Note. This dataset is from 1978, recently, we are facing shortage of professors worldwide. With current situation, we should be careful with this conclusion.

4.4. Aristotle and the skilled questioner

Even thought code interpreter can do miracles, it seems to have a weakness: you need to know what to ask. This is similar to what Aristotle believed when teaching someone: you need to be a skilled questioner. Even though if you ask, it will do everything automatically, you need to know what to ask for getting information from the dataset you have. This is just possible if you are already minimally familiar with *statistical inference*. Thus, we believe this tool will be most efficient at the hand of data scientists minimally trained on statistical inference, and related areas that make up data science.

Aristotle believed that knowledge is not something that can be taught, but rather something that is already present within us. He believed that the role of the teacher is to help students bring this knowledge to the surface through a process of questioning and reflection. In other words, Aristotle believed that everyone has the potential to be a philosopher, but that we need to be skilled questioners in order to uncover the knowledge that is already within us. The same way, with the current tool, anyone can be a data scientist, as long as they get a minimal training in statistical inference.

This idea is closely related to Aristotle's concept of epagoge, or "induction." According to Aristotle, induction is the process of moving from particular instances to general principles. By asking questions and reflecting on our experiences, we can begin to identify patterns and make generalizations about the world around us .

In summary, Aristotle believed that knowledge is not something that can be taught, but rather something that is already present within us. By asking questions and reflecting on our experiences, we can begin to uncover this knowledge and develop a deeper understanding of the world around us.

4.5. Related discussions

4.5.1. Other options

Although I have discussed openAI API and created my own app calledSheetChat using this approach (see SM), it is becoming a standard very fast to use artificial intelligence alongside spreadsheets, data science guided by artificial intelligence. It is already possible to find extensions for browser that will do the same trick, also add-ins for spreadsheets. Thus, even though I have brought to attention the coder interpreter, it is becoming a common feature for data scientists to have around artificial intelligence powered tools on their data science analysis tasks.

4.5.2. Limitations and potential challenges

There are limitations that should be considered that also extents to other similar tools; in addition to limitations from the tool you are using, which may vary.

One limitation is that I have not double checked the calculations I have reported. I assume that this is correct. There is no evidence that the model did something wrong, such as running the wrong tests or hallucinating. However, for a scientific publication, where the stakes are high, it is advisable to double-check the calculations manually (see SM). Initially, I thought to include double checking the calculations in Excel; the issue is that each case by itself has several steps to arrive at the final result. The nice point about the tool is that it makes a data science task precisely in seconds, which would take hours, or days. Until we have more cases of success, it is strongly recommended not to trust blindly on this tool. A possible promising future work is testing several cases, one by one, and reporting whether the model makes mistakes, and where, what are the commonest mistakes it generally does.

Although the model is generally very smart, even for fixing errors on the dataset automatically, it can fail. One example is presented on the demos, where the model insisted to plot a graph, with no information, a blank graph. Even after several attempts, the model was unable to plot the graph with the data. The issue was localized on a CSV, it is not something common. It is possible that the format used to generate the CSV confused the AI, even though generally it can spot those issues and automatically fix them. The demo was deployed on our app here.

4.5.3. Avoid internal biases and ideologies when interpreting datasets

Daniel Kahneman in Thinking, Fast and Slow[35] brought to attention that we humans have several biases that influence our thinking. Later, Daniel Kahneman and colleagues [36] also added noise to flaws in human thinking. Our goal here is not to make a complete discussion; it is just to add to the discussion.

Since the tool is based on a LLM, it has access to several different perspectives. Thus, any bias related to giving too much value to the researcher knowledge and background may be mitigated. AI models are good at integrating a big set of different sources.

Although it has been shown in some studies that LLMs are not free from ideologies, they are strongly vulnerable to their datasets during training, they are susceptible to different biases, as I see it. Their biases will be more "crowd biases", whereas humans will be strongly biased based on local and individual biases.

Of course, those are all speculations and more studies should be done in the future. It would be very interesting and valuable to enrich the literature on biases on LLMs as they become more and more used. It is hard to compare currently because the literature on humans is rich, but for LLMs, they are almost no existent.

4.5.4. Boundaries and limitations of the OpenAI API's capabilities in statistical analysis

The application of LLMs in data science is new. Thus, care should be taken. As highlighted before, the greatest risk is that the model generates a wrong analysis, maybe using a wrong library. One solution for that is asking the code interpreter more information on the analysis, such as "which library did you use for the analysis?". An alternative is double checking the methods used, for instance, by actually doing a careful search on textbooks, on the internet. As a last resource, doing all the calculations "by hand", using traditional approaches. With time and as those tools spread out relatively fast, more and more cases will become public. This means also their weaknesses and strengths. Since chatGPT was launched, we already have some clarity about its limitations and strengths now, even though we are still learning everyday.

For finding scenarios where the API might struggle or produce inaccuracies to guide users in its application, it may need time. Nevertheless, I have attached one, where it could not produce a graph from the dataset. Apparently, it had problems with the date format used for the historical records of COVID. As we use this tools, it will become clear its limitations. So far, it had succeeded on most of the tasks given. For example, in a parallel study, it produced a heat map from a CSV file, using an actual map, with the names of small towns, a Google Map with the accuracy of snake species from a CSV file. It was actually impressive, just with a CVS file and a generic request, with a generic description of the dataset. We were studying the geographical distributions of snake species.

4.5.5. Accuracy and Reliability

One limitation of the current study is that I have not made a comparison with traditional approaches. The coder interpreter uses classical Python data science libraries. Therefore, what we need to monitor on the future in another research is whether it is picking the right libraries, once it picked the right library, it is no longer a problem from the coder interpreter, it is an external library, used also by traditional data science. The possible testing here for proving accuracy and reliability is

whether it is picking the right tools and using the results properly to arrive at the conclusion. I have actually done that on another scenario, using the same API (except it was not coder interpreter, but the API is the same) [6][7].

One validation that I have performed is to search the literature to confirm the finding. If the model had used the wrong libraries or had done the calculations wrongly, the literature would have pointed that out, and it did not. For instance, in the case of diabetes being more present in men, this is a well-known result on the literature. Thus, one way to double check the results is by comparing them with those in the literature. In case the literature is non-existent, it would be necessary to redo the calculations. It is always possible to ask the tool details about the calculations done, such as which tool was used, and make the judgment based on that.

One possible way to increase Accuracy and Reliability is providing your own tools: it can be passed as a tool box. It is possible to either store those functions on their server, with a charge, or store it locally. In this scenario, code expertise is required. Once set, others can use it. SheetChat is an example, even though I am now using just their default/standard tools, I can create a toolbox and make available to be called. It will be available to the user; see [6][7] for examples on how to do it.

4.5.6. User Interface and Accessibility

The coder interpreter is provided by API, thus, one must know how to code minimally. However, it is possible to build interfaces, such as SheetChat. With the interface, it is possible to do data science without any (strong) expertise in either coding or data science. Their playground is also friendly; it is not hard to use after a simple and basic setup. As for all the LLMs from openAI API, they are straightforward to use: text in text out. One must do a basic setup before using their playground, if preferred (Fig. 3).

4.5.7. Potential Biases or Limitations

Coder interpreter picks the tool it will use, and it could happen over-reliance on certain libraries or methods. The tool used will certainly influence the final result, the final conclusions. One solution for that is asking the coder interpreter more details on the analysis, and ask it directly to use say a specific approach. When the approach is not available, it must be provided as tools. Learn more about this approach on my other publications [6][7].

4.5.8. Use Cases and Examples

I have provided two SMs: one using openAI API playground, and other using SheetChat. They are examples such as on diabetes. SheetChat has a section of demos, which I am planning to enrich in the future. Except for issues on reading the dataset, it seems there is no restrictions on the use cases. When limits are reached, one can always add tools for fulfilling gaps. SheetChat has this possible future, it was coded in TypeScript.

4.5.9. Future Research and Improvements

Upgrades can be done either by adjusting the assistant description, called*prompt engineering*; or by providing extra tools.

It is also possible to direct the tools by asking it directly, as a chat message. It can be done by the user with time, as the limitations are found. It is a good practice to use the defaults/standard and just add new features if needed, which may be totally dependent from the user's applications. The environment seems very rich for standard analysis generally done by data scientists. They are using well-established libraries in Python, Python is well-known for its central role in data science as open-source. Thus, it is safe to assume that it will suffice to most of the cases one may face.

4.5.10. Would humans be replaced?

No. there are no evidence that human experts will be replace, what will most likely happen is an automation of basic tasks, that are already automated by libraries. Coder Interpreter is using Python libraries, what it is doing is deciding which tools to use, given a rich menu, and using its language capabilities as LLMs to interpret the results, but it does not replace humans as final end. What I see is a new way to make data science, something that already happened when those libraries used started to gain momentum. It is just a step forward towards something already in course: making repetitive, but important, research steps into straightforward tasks, without losing scientific rigours. Most data scientists are not necessarily computers and mathematical sorcerers. Thus, the fact that they can perform important statistical analysis without needing a multidisciplinary team, or even, needing to know the basics all the time is something to celebrate. Of course, we still need human's expertise and guidance, deciding what to prioritize. Even, in some cases, the correct conclusions. Human will not be replace, it will just have more energy for asking the big questions. Coder interpreter is a productive tool, not a artificial general intelligence.

Automation in data science aims to facilitate and transform the work of data scientists, not to replace them[37].

5. Conclusion

We have explored a new tool from openAI called coder interpreter. This tool is able to find knowledge from datasets. We have shown that the tool can make basic statistical inference automatically and, with guidance, even go further, like the Aristotle method: You ask to take knowledge out of a dataset. This tool can be particularly interesting to data scientists already familiar with statistical inference. We have in a problem from medicine, making the tool particularly interesting to medical doctors: once they have a basics on statistics, they can perform statistical analysis on their datasets effortless, enforcing evidence-based medicine.

Acknowledgements

This article was published under Open Access Review on Qeios.com. I would like to openly acknowledge the following important reviewers, who helped me to improve: Adriano Bressane (Universidade Estadual Paulista); Arash Heidari (Institute of Electrical and Electronics Engineers (IEEE)).

References

1. [a, b]*HAO, K. The chaos inside OpenAI – Sam Altman, Elon Musk, and existential risk explained — Karen Hao. Big Think [YouTube Channel], 2023. Accessed on 2 Dec 2023. Disponível em: ⟨https://www.youtube.com/watch?v=O9sLCp2Jq74&t=3333s⟩.*
2. [a, b]*WOLFRAM, S. What Is ChatGPT Doing... and Why Does It Work? 2023. https://writings.stephenwolfram.com/2023/02/whatis-chatgpt-doing-and-why-does-it-work/.*
3. [^]*PIRES, J. G. O mercado da criatividade: Regulamentação da profissão de pesquisador acadêmico e científico no Brasil. 2023. Disponível em: ⟨https://www.amazon.com.br/mercado-criatividade-Regulamenta% C3%A7%C3%A3o-pesquisador-cient%C3%ADfico-ebook/dp/B09TKRX5PW⟩.*
4. [a, b, c]*BULLARD, K. M. et al. Prevalence of diagnosed diabetes in adults by diabetes type — united states, 2016. Morbidity and Mortality Weekly Report, US Department of Health and Human Services, Centers for Disease Control and Prevention, v. 67, n. 12, p. 359, 2018.*
5. [a, b, c, d, e, f, g, h, i, j]*TITUS, A. J. Nhanes-gpt: Large language models (llms) and the future of biostatistics. medRxiv, Cold Spring Harbor Laboratory Press, 2023. Disponível em: ⟨https://www.medrxiv.org/content/early/2023/12/15/2023.12.13.23299830⟩.*
6. [a, b, c, d, e]*PIRES, P. J. G. Robodoc: a conversational-ai based app for medical conversations. Qeios, 2024.*
7. [a, b, c, d, e]*PIRES, P. J. G. Snakechat: a conversational-ai based app for snake classification. Qeios, 2024.*
8. [a, b, c]*PIRES, J. G. Machine learning in medicine using javascript: building web apps using tensorflow.js for interpreting biomedical datasets. medRxiv, Cold Spring Harbor Laboratory Press, 2023. Disponível em: ⟨https://www.medrxiv.org/content/early/2023/12/21/2023.06.21.23291717⟩.*
9. [^]*AUTOMATED Machine Learning - Methods, Systems, Challenges. In: AUTOMATED Machine Learning. [s.n.], 2019. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:68018984⟩.*
10. [a, b, c]*HOLLMANN, N.; MULLER, S.; HUTTER, F. Large language models for automated data science: Introducing caafe for context-aware automated feature engineering. In:. [s.n.], 2023. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:258547322⟩.*
11. [a, b, c, d, e]*NOEVER, D. A.; MCKEE, F. Numeracy from literacy: Data science as an emergent skill from large language models. ArXiv, abs/2301.13382, 2023. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:256416333⟩.*
12. [a, b, c]*MONTGOMERY, D. C.; RUNGER, G. C. Applied Statistics and Probability for Engineers. [S.l.]: John Wiley & Sons, 2010.*
13. [^]*KAHNEMAN, D. Thinking, Fast and Slow. [S.l.]: Farrar, Straus and Giroux, 2011.*
14. [^]*IAMARINO, A. Por que ideologias podem emburrecer. 2023. Accessed on 27 Jan 2024. Disponível em: ⟨https://www.youtube.com/watch?v=NHAOI54U-Aw&t=411s⟩.*
15. [^]*LEUNG, T. I. et al. Can ai mitigate bias in writing letters of recommendation? JMIR Medical Education, JMIR Publications Inc., Toronto, Canada, v. 9, n. 1, p. e51494, 2023.*
16. [^]*MAY, J. Bias in science: natural and social. Synthese, Springer, v. 199, n. 8, p. 3345–3366, 2021.*
17. [^]*TODAY, P. Recognizing PoliticallyBiased Social Science. 2021. ⟨https://www.psychologytoday.com/us/blog/psych-*

unseen/202112/recognizing-politically-biased-social-science).

18. ^*ACADEMY, E. Dealing With Bias in Academic Research. 2022. (https://www.enago.com/academy/dealing-with-bias-in-academic-research/)*.
19. ^*LEARNING, S. N. How bias affects scientific research. (https://www.sciencenewsforstudents.org/article/how-bias-affects-scientific-research)*.
20. ^*UNIVERSITY, S. Ideological Bias and Trust in Information Sources. (https://web.stanford.edu/~gentzkow/research/biasmeas/biasmeas.pdf)*.
21. ^*GALLEGOS, I. O. et al. Bias and fairness in large language models: A survey. arXiv preprint arXiv:2109.05252, 2021.*
22. ^*LI, Y. et al. Generative language models exhibit social identity biases. arXiv preprint arXiv:2104.04451, 2021.*
23. ^*SUN, H. et al. Aligning with whom? large language models have gender and racial biases in subjective nlp tasks. arXiv preprint arXiv:2104.07693, 2021.*
24. ^*Kumar, A.; Kumar, S. Python for data analytics, scientific and technical applications. In: 2019 Amity International Conference on Artificial Intelligence (AICAI). [S.l.: s.n.], 2019. p. 1–5.*
25. ^*Kumar, M.; Kumar, S. An overview and comparison of free python libraries for data mining and big data analysis. In: 2018 3rd International Conference on Computing Methodologies and Communication (ICCMC). [S.l.: s.n.], 2018. p. 1–5.*
26. a, b, c*HEARN, E. Is proprietary software better than opensource? Built In, June 2020. Disponível em: (https://builtin.com/data-science/open-source-proprietary-software)*.
27. ^*CASTILLO, L. Open Source Software Statistics [Fresh Research]. 2023. Disponível em: (https://blog.gitnux.com/open-source-software-statistics/)*.
28. ^*KAUTZKY-WILLER, A.; LEUTNER, M.; HARREITER, J. Sex differences in type 2 diabetes. Diabetologia, Springer Link, v. 66, n. 1, p. 986–1002, 2023.*
29. ^*CIARAMBINO, T. et al. Influence of gender in diabetes mellitus and its complication. International Journal of Molecular Sciences, v. 23, n. 16, p. 8850, 2022.*
30. ^*BULLARD, K. M. et al. Prevalence of diagnosed diabetes in adults by diabetes type — united states, 2016. Morbidity and Mortality Weekly Report, US Department of Health and Human Services, Centers for Disease Control and Prevention, v. 67, n. 12, p. 359, 2018.*
31. a, b*PIRES, J. G. Discussing a multiple regression model. 2022. Disponível em: (https://medium.com/computational-thinking-how-computers-think-decide/discussing-a-multiple-regression-model-c1f45ce52211)*.
32. a, b, c*NIELSEN, S. C. E. D.; BILESCHI, S. M.*
33. a, b*PIRES, J. G. Computational Thinking: How computers think, decide and learn, when human limits start and computers champ, vol. 1. 2022. Disponível em: (https://books.google.com/books/about/Computational_Thinking_How_computers_thi.html?id=WfV6EAAAQBAJ)*.
34. a, b*KAHNEMAN, D.; SIBONY, O.; SUNSTEIN, C. R. Noise: A Flaw in Human Judgment. [S.l.]: William Collins, 2021.*
35. ^*KAHNEMAN, D. Thinking, fast and slow. [S.l.]: Farrar, Straus and Giroux, 2011.*
36. ^*SUNSTEIN, D. K. C. R.; SIBONY, O. Noise: A Flaw in Human Judgment. [S.l.]: Little, Brown Spark, 2021.*
37. ^*BIE, T. D. et al. Automating data science. Communications of the ACM, v. 65, p. 76 – 87, 2021. Disponível em:*

(https://api.semanticscholar.org/CorpusID:247059314).