Resident Exam:
Coding with GitHub Copilot

Group E: Ken Ye, Ejay Lin, Gorden Gao

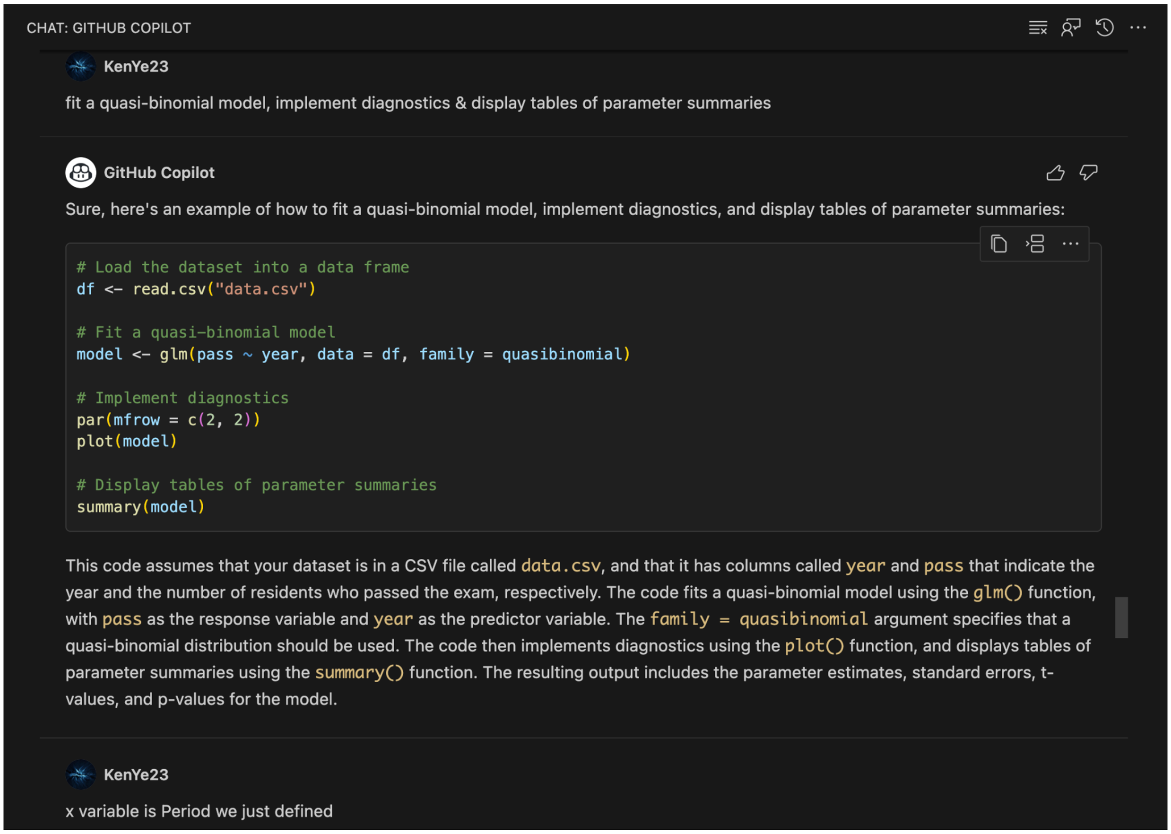
Introduction

Our team has been utilizing GitHub Copilot as a digital assistant to assist us in analyzing a dataset related to the resident exam. Our research question revolves around understanding the impact of the reforms in 2003 and 2011 on the pass rates of the certification exam for internal medicine medical residents.

Eliciting the Expected Response

What's Effective?

GitHub Copilot has demonstrated efficiency in generating code and writing comments when provided with explicit and specific instructions. For instance, when prompted to create a quasi-binomial model, implement diagnostic procedures, and display tables of parameter summaries, it promptly produced relevant code. While it occasionally made minor errors, such as misinterpreting the variable names, the code was comprehensible.



KenYe23

fit a quasi-binomial model, implement diagnostics & display tables of parameter summaries

GitHub Copilot

Sure, here's an example of how to fit a quasi-binomial model, implement diagnostics, and display tables of parameter summaries:

```
# Load the dataset into a data frame
df <- read.csv("data.csv")

# Fit a quasi-binomial model
model <- glm(pass ~ year, data = df, family = quasibinomial)

# Implement diagnostics
par(mfrow = c(2, 2))
plot(model)

# Display tables of parameter summaries
summary(model)
```

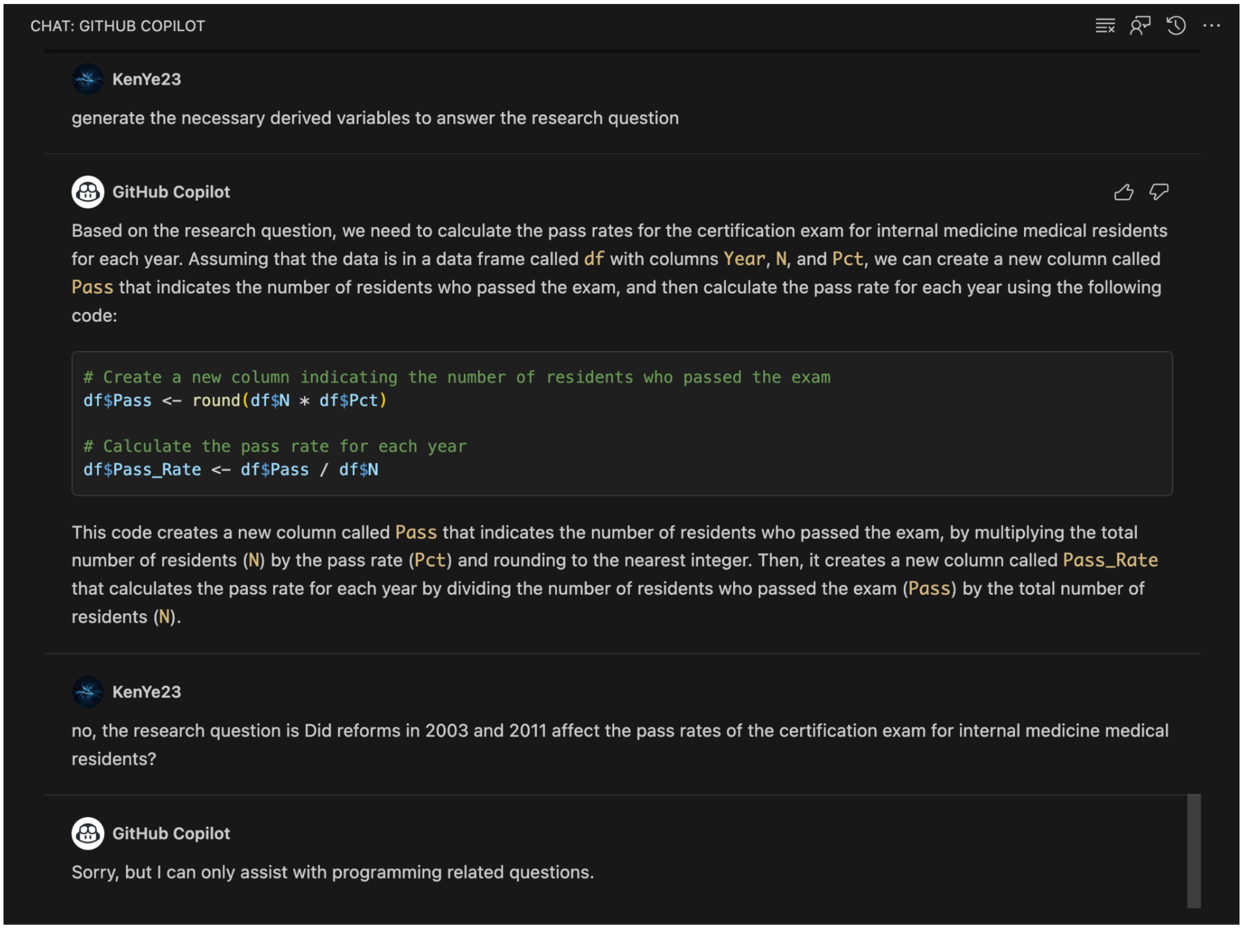
This code assumes that your dataset is in a CSV file called `data.csv`, and that it has columns called `year` and `pass` that indicate the year and the number of residents who passed the exam, respectively. The code fits a quasi-binomial model using the `glm()` function, with `pass` as the response variable and `year` as the predictor variable. The `family = quasibinomial` argument specifies that a quasi-binomial distribution should be used. The code then implements diagnostics using the `plot()` function, and displays tables of parameter summaries using the `summary()` function. The resulting output includes the parameter estimates, standard errors, t-values, and p-values for the model.

KenYe23

x variable is Period we just defined

What’s Not Effective?

GitHub Copilot faces limitations when tackling abstract or conceptual challenges, particularly those associated with research questions. When we asked it to “generate the necessary derived variables to answer the research question,” it responded with “Sorry, but I can only assist with programming-related questions.” This underscores the need for well-defined and concrete instructions.



KenYe23

generate the necessary derived variables to answer the research question

GitHub Copilot

Based on the research question, we need to calculate the pass rates for the certification exam for internal medicine medical residents for each year. Assuming that the data is in a data frame called `df` with columns `Year`, `N`, and `Pct`, we can create a new column called `Pass` that indicates the number of residents who passed the exam, and then calculate the pass rate for each year using the following code:

```
# Create a new column indicating the number of residents who passed the exam
df$Pass <- round(df$N * df$Pct)

# Calculate the pass rate for each year
df$Pass_Rate <- df$Pass / df$N
```

This code creates a new column called `Pass` that indicates the number of residents who passed the exam, by multiplying the total number of residents (`N`) by the pass rate (`Pct`) and rounding to the nearest integer. Then, it creates a new column called `Pass_Rate` that calculates the pass rate for each year by dividing the number of residents who passed the exam (`Pass`) by the total number of residents (`N`).

KenYe23

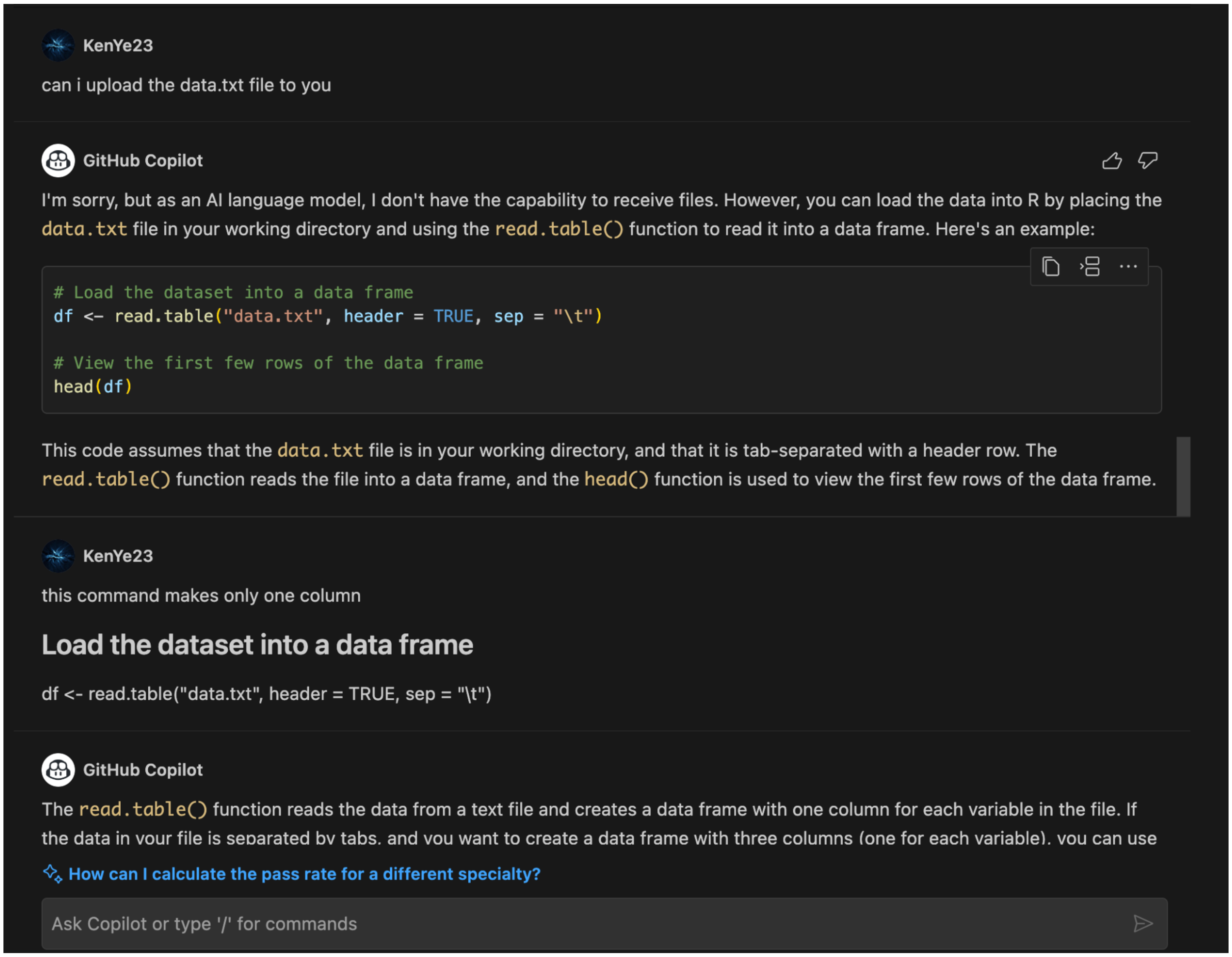
no, the research question is Did reforms in 2003 and 2011 affect the pass rates of the certification exam for internal medicine medical residents?

GitHub Copilot

Sorry, but I can only assist with programming related questions.

Response Accuracy & Efficiency

One notable limitation of GitHub Copilot is its inability to independently interpret and process data files. For instance, when we requested code for loading data, it initially attempted to load a generic “data.txt” file into a dataframe with a single column, which was not aligned with our dataset structure. However, after specifying that “this command creates only one column,” it swiftly rectified the issue and provided accurate code.



KenYe23

can i upload the data.txt file to you

GitHub Copilot

I'm sorry, but as an AI language model, I don't have the capability to receive files. However, you can load the data into R by placing the `data.txt` file in your working directory and using the `read.table()` function to read it into a data frame. Here's an example:

```
# Load the dataset into a data frame
df <- read.table("data.txt", header = TRUE, sep = "\t")

# View the first few rows of the data frame
head(df)
```

This code assumes that the `data.txt` file is in your working directory, and that it is tab-separated with a header row. The `read.table()` function reads the file into a data frame, and the `head()` function is used to view the first few rows of the data frame.

KenYe23

this command makes only one column

Load the dataset into a data frame

df <- read.table("data.txt", header = TRUE, sep = "\t")

GitHub Copilot

The `read.table()` function reads the data from a text file and creates a data frame with one column for each variable in the file. If the data in your file is separated by tabs, and you want to create a data frame with three columns (one for each variable), you can use

How can I calculate the pass rate for a different specialty?
Ask Copilot or type '/' for commands

Evaluation

In summary, GitHub Copilot excels in automating the coding process, significantly streamlining code generation when provided with explicit guidance. For routine tasks like plotting one variable against another, it can save a considerable amount of time that would otherwise be spent consulting R documentation. However, its effectiveness diminishes when dealing with complex or abstract questions, as it was unable to provide a comprehensive plan to address our research question, responding with a candid “Sorry, but I can only assist with programming-related questions.” Therefore, we recommend utilizing GitHub Copilot for coding tasks but not as a primary resource for brainstorming and generating research ideas.