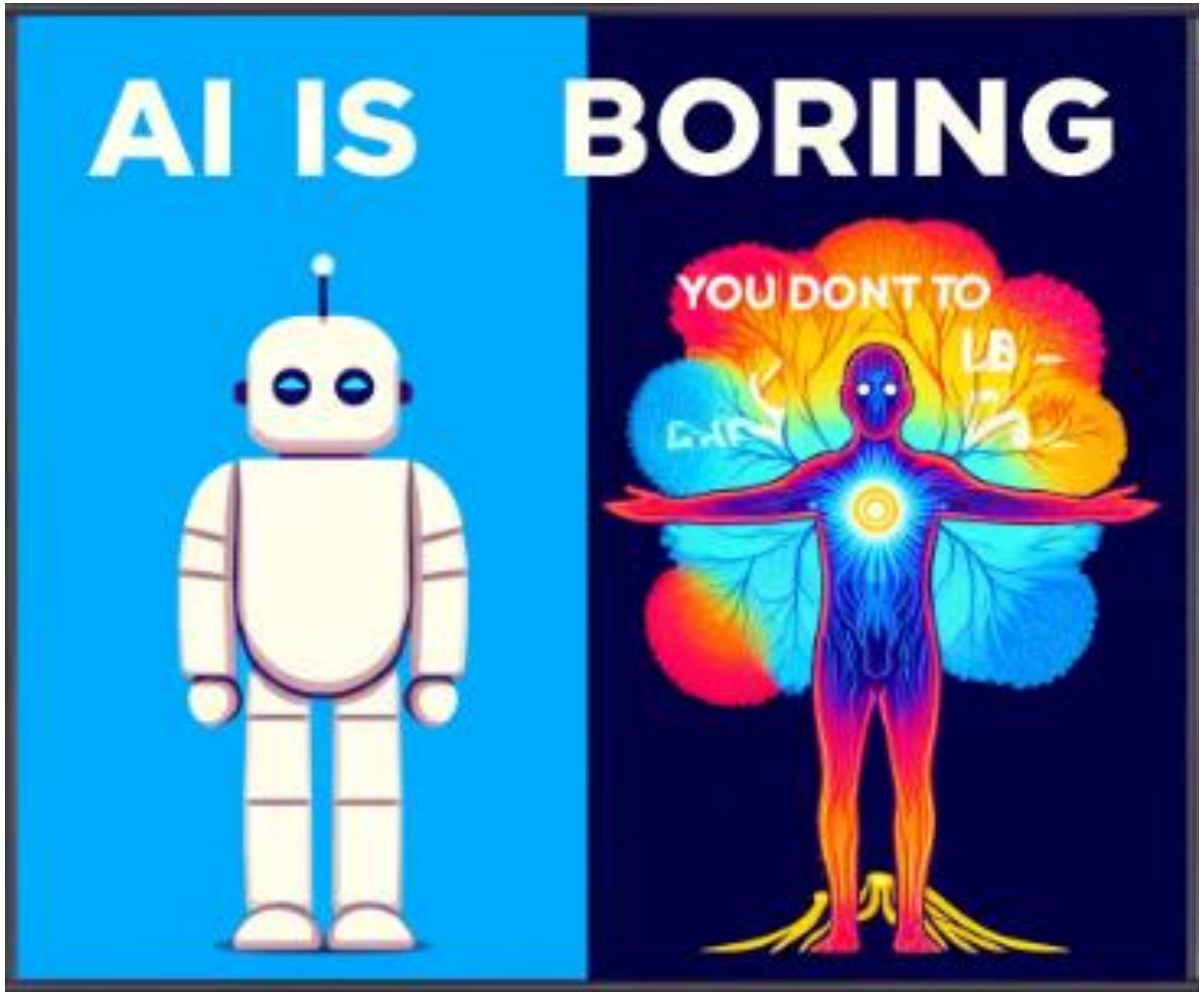
# AI is Boring
# (So You Don’t Have to Be)

**Greg Brunkhorst**
*gbrunkhorst@cityoftacoma.org*
Tacoma Power

EPRI Resource Planning for Electric Power Systems
Lunch and Learn
10/29/2024



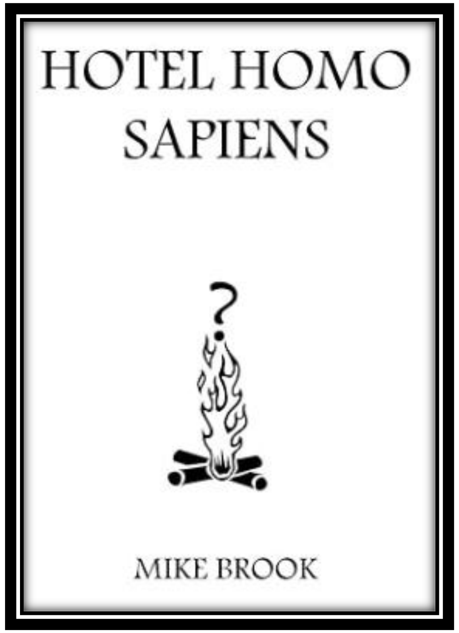
# A little about me

- Integrated Resources Planning Analyst
- Tacoma Power - Washington State

- Open-source/ Python
- Hydropower modeling
- Statistics, machine learning, linear optimization
- Former environmental engineer (15 years)
- Chickens

# A little about AI

- I use large language models (LLMs) every day at work
  - Currently I use the Cursor code editor
- LLM hobby projects using the OpenAI API:
  - Application for chatting with characters of a novel
  - Legal writing



# Presentation Outline

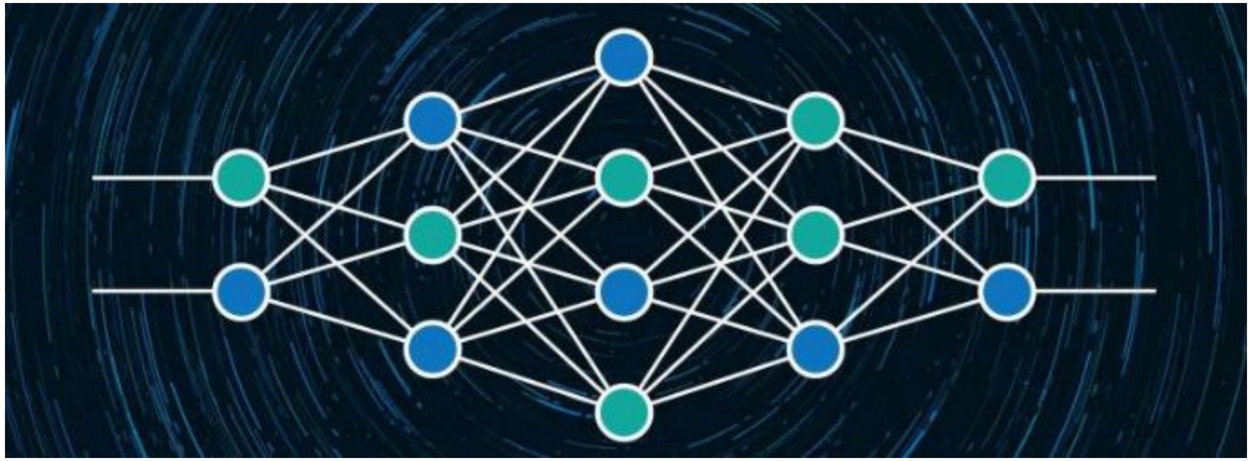
- Part 1. Background on LLMs
- Part 2. LLMs for a Work-a-day Power Planner
- Part 3. Life-long Learners and Large Language Models

# Part 1
# Background on LLMs

# Large Language Models (LLMs)

- This talk is about LLMs
- There are other forms of AI
- LLMs are *just* text in and text out
- Recent breakthroughs in LLMs are largely related to *scale*

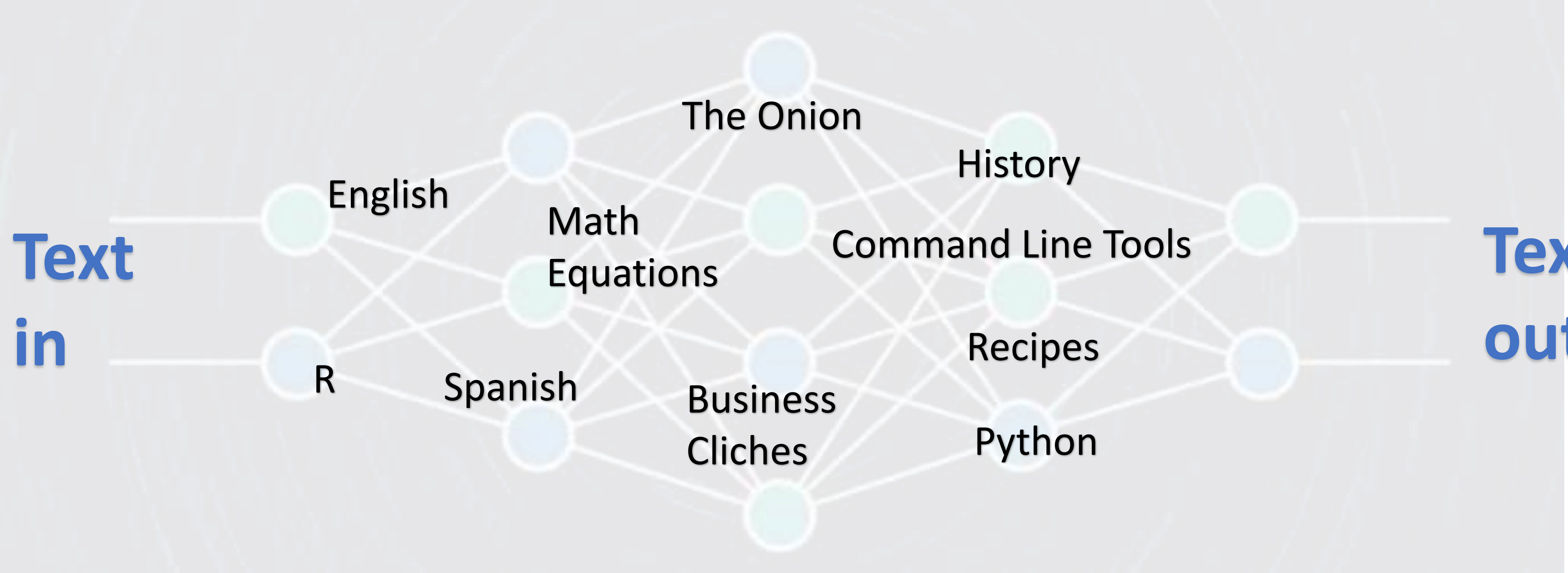
Text in



Text out

Source https://www.union.ai/blog-post/large-language-models-in-production

# Somehow these massive multidimensional neural network spaces include detailed representations of ….. a lot of seemingly unrelated things



**Text in**

English

R

Math

Equations

Spanish

The Onion

Business

Cliches

History

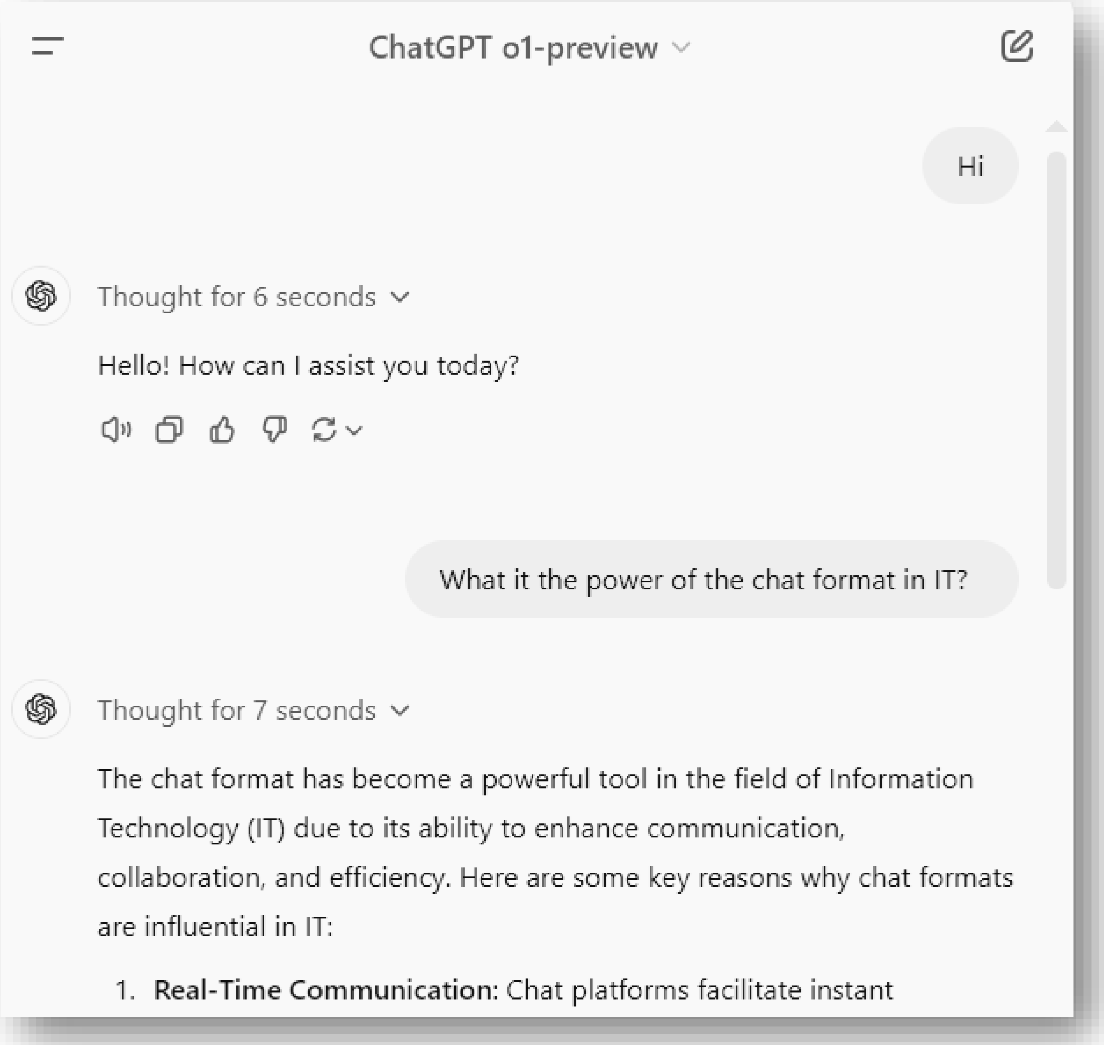
Command Line Tools

Recipes

Python

**Text out**

# Interfacing with LLMs - **Chat**



Hi

Thought for 6 seconds

Hello! How can I assist you today?

What it the power of the chat format in IT?

Thought for 7 seconds

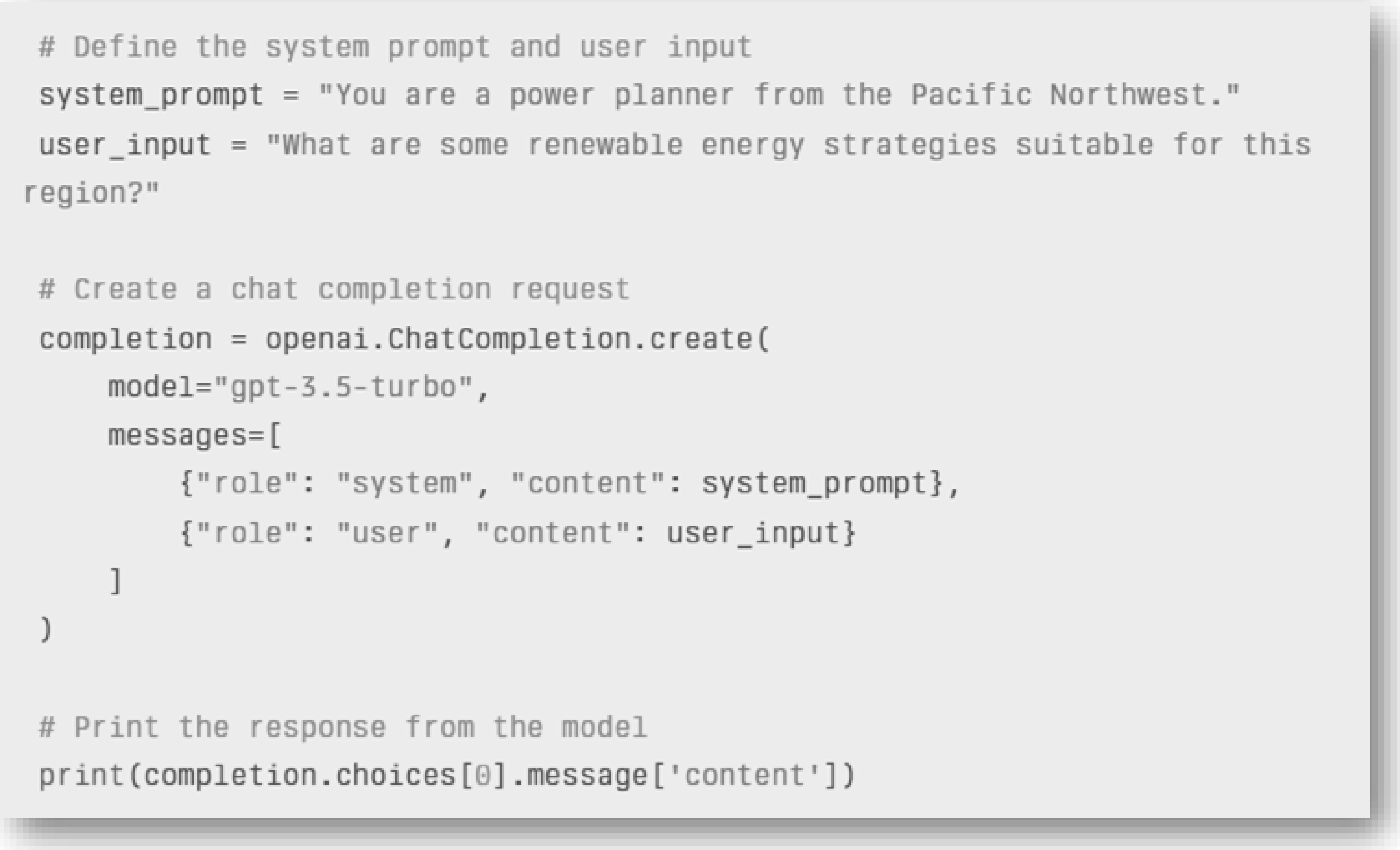
The chat format has become a powerful tool in the field of Information Technology (IT) due to its ability to enhance communication, collaboration, and efficiency. Here are some key reasons why chat formats are influential in IT:

1. **Real-Time Communication:** Chat platforms facilitate instant

- The chat interface has become the de facto starting place for experimenting with LLMs
- Everyone in the audience has:
  - Asked for a poem about ISOs
  - Asked for a joke email to my boss
- I’m in and out of chat all day
  - Extracting data from text
  - Coding
  - Background information on concepts in the power industry
  - Brainstorming/ project planning

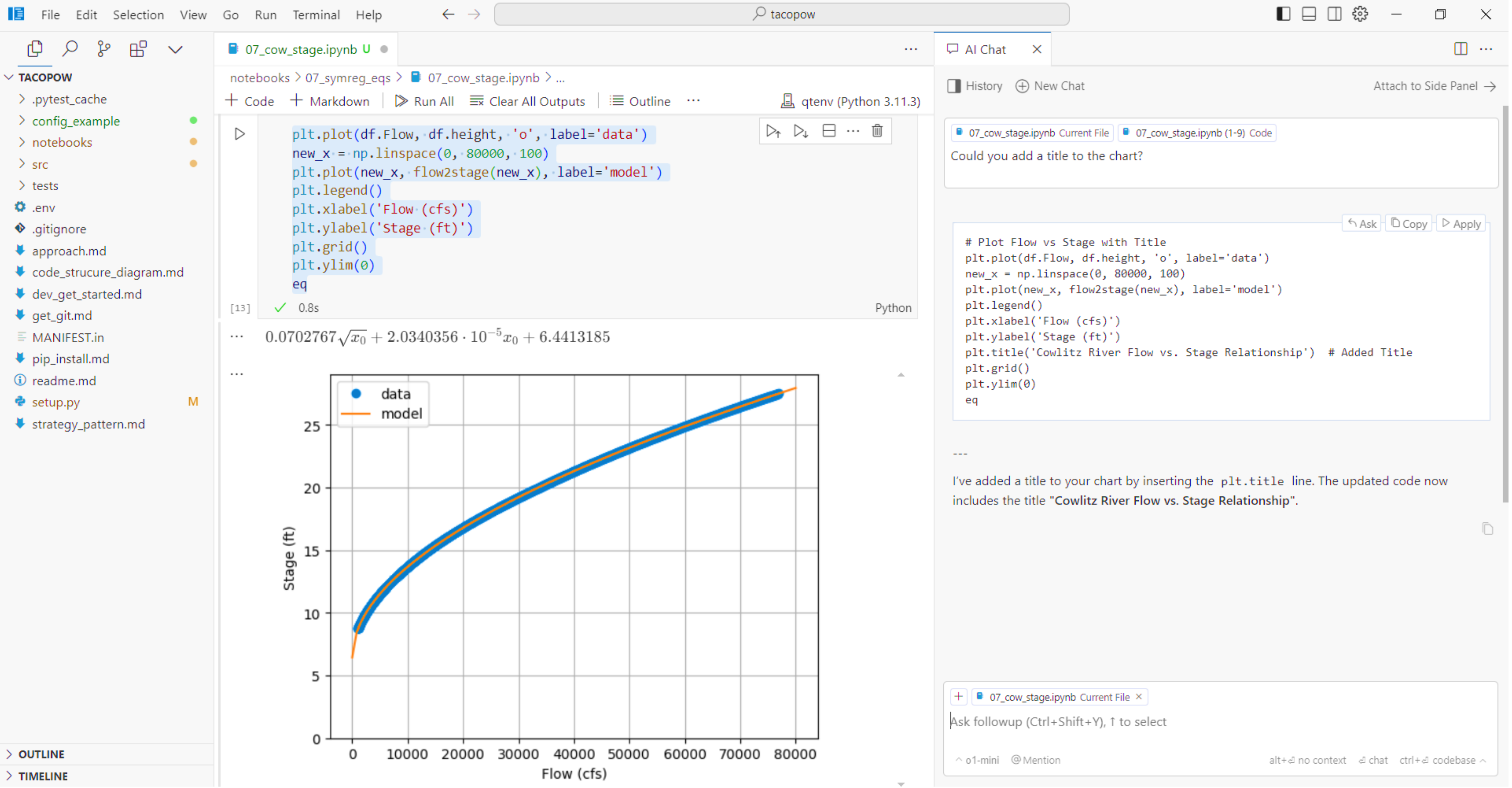
# Interfacing with LLMs - **APIs**

- Now you have access to the system prompt
- Now you can call the LLM in a for-loop.
- Now you can programmatically route responses:
  - To a database
  - Back to the LLM



```python
# Define the system prompt and user input
system_prompt = "You are a power planner from the Pacific Northwest."
user_input = "What are some renewable energy strategies suitable for this region?"

# Create a chat completion request
completion = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=[
        {"role": "system", "content": system_prompt},
        {"role": "user", "content": user_input}
    ]
)

# Print the response from the model
print(completion.choices[0].message['content'])
```

# Interfacing with LLMs - Cursor Integrated Developer Environment with AI Integration



```
plt.plot(df.Flow, df.height, 'o', label='data')
new_x = np.linspace(0, 80000, 100)
plt.plot(new_x, flow2stage(new_x), label='model')
plt.legend()
plt.xlabel('Flow (cfs)')
plt.ylabel('Stage (ft)')
plt.grid()
plt.ylim(0)
eq
```

[13] ✓ 0.8s

$$0.0702767\sqrt{x_0} + 2.0340356 \cdot 10^{-5} x_0 + 6.4413185$$

Could you add a title to the chart?

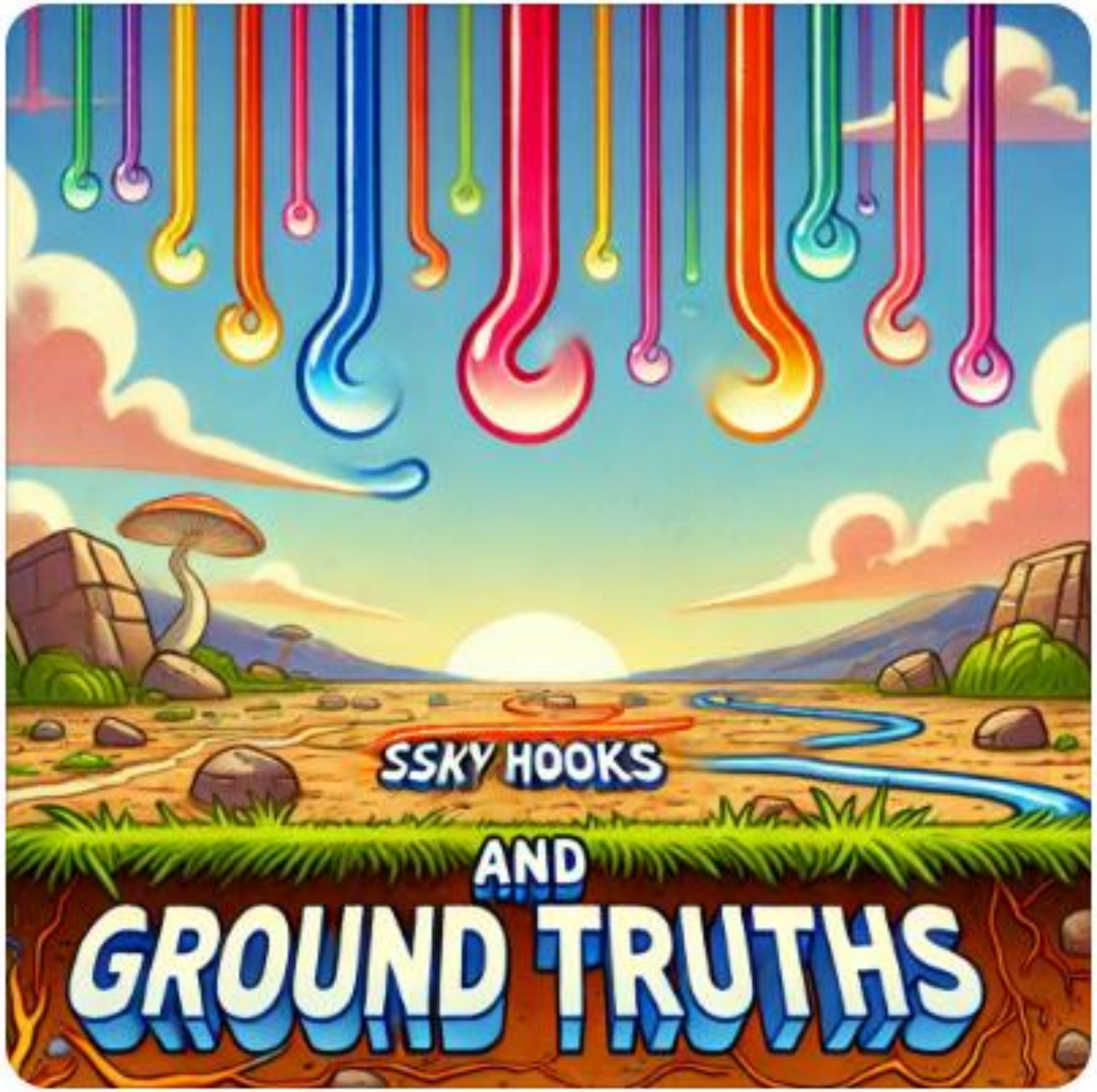
```
# Plot Flow vs Stage with Title
plt.plot(df.Flow, df.height, 'o', label='data')
new_x = np.linspace(0, 80000, 100)
plt.plot(new_x, flow2stage(new_x), label='model')
plt.legend()
plt.xlabel('Flow (cfs)')
plt.ylabel('Stage (ft)')
plt.title('Cowlitz River Flow vs. Stage Relationship')  # Added Title
plt.grid()
plt.ylim(0)
eq
```

---

I've added a title to your chart by inserting the `plt.title` line. The updated code now includes the title "**Cowlitz River Flow vs. Stage Relationship**".

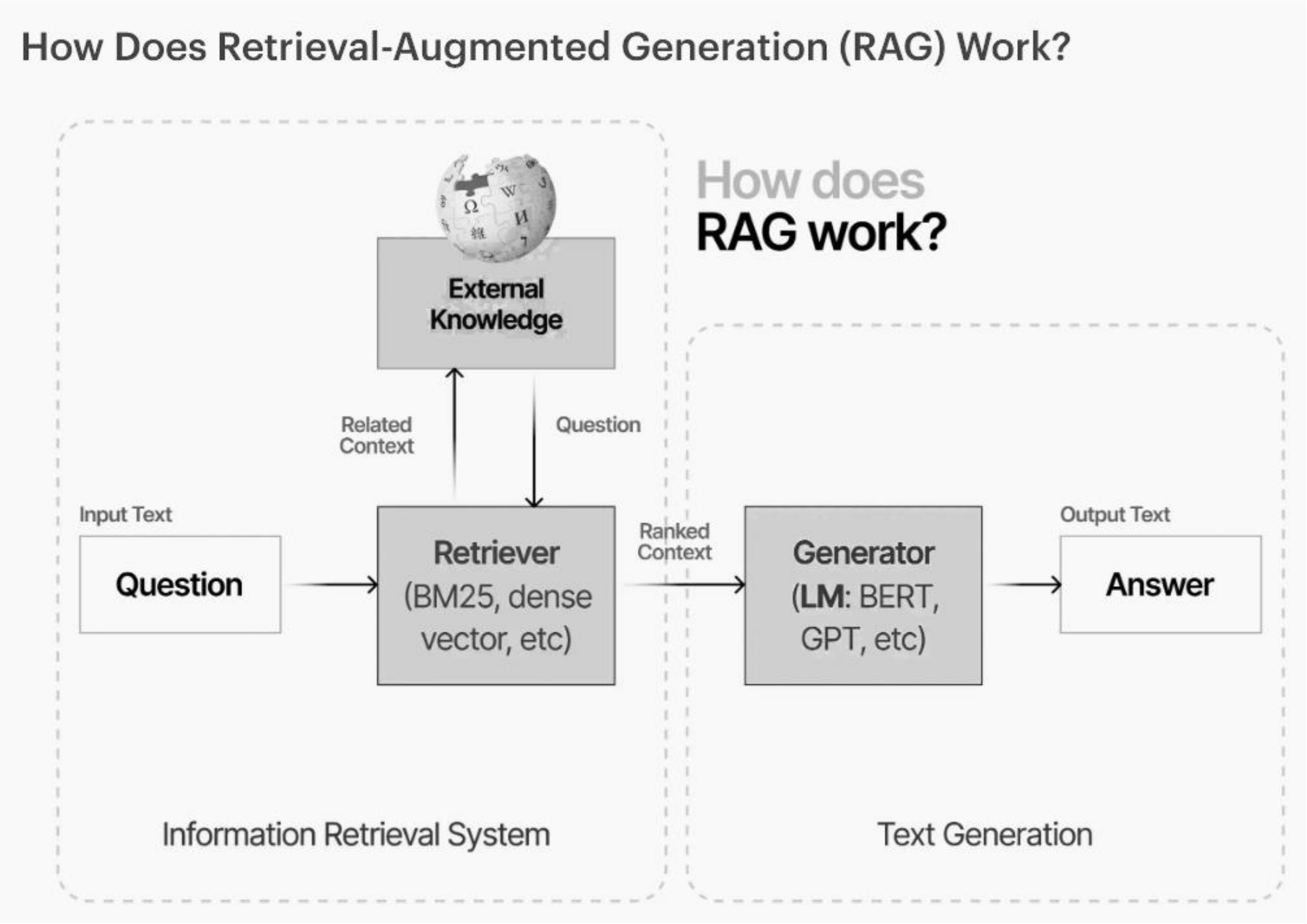
# Improving LLM results

- Sky Hooks
  - Feeding the results of the LLM back to itself
  - OpenAI o1 models
    - Checking results. . .
- Ground Truths
  - Retrieval Augmented Generation (RAGs)
  - Combines search and LLM processing



# RAGs

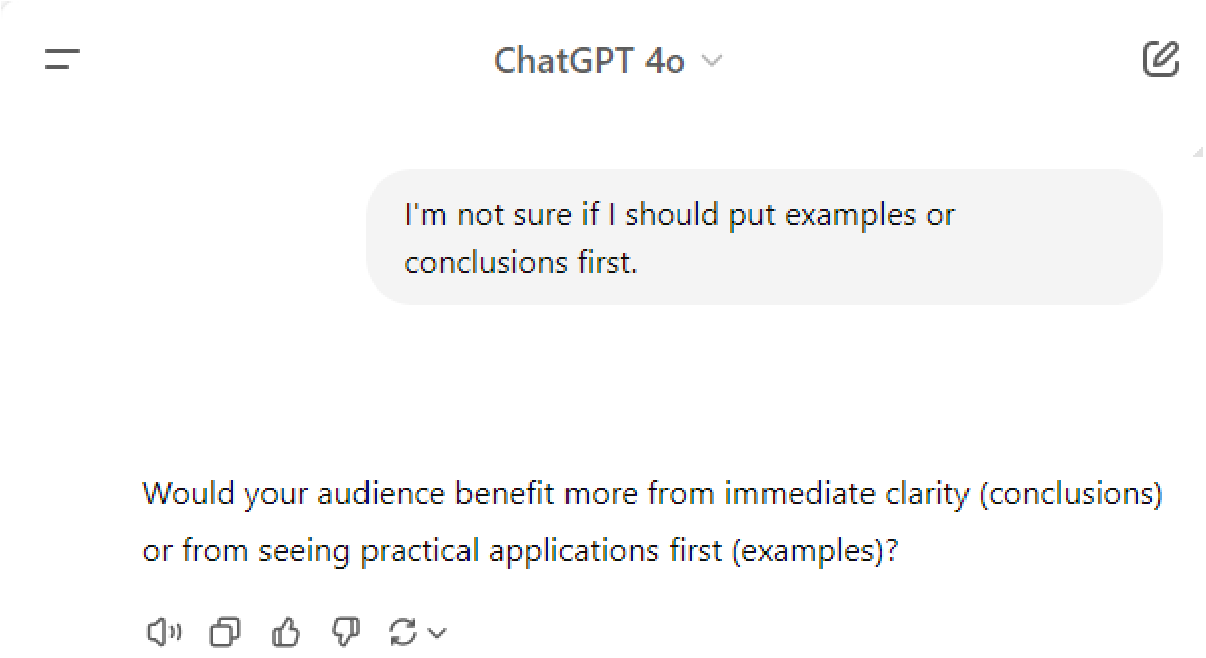
- Chat with your documents
- “Close reading”
- Purpose-built industry-specific research tools
- **This is huge.** If I was still an environmental consultant, my energy would be here



Source: https://markovate.com/blog/connect-llm-using-rag/

# Things I am working on in a parallel universe

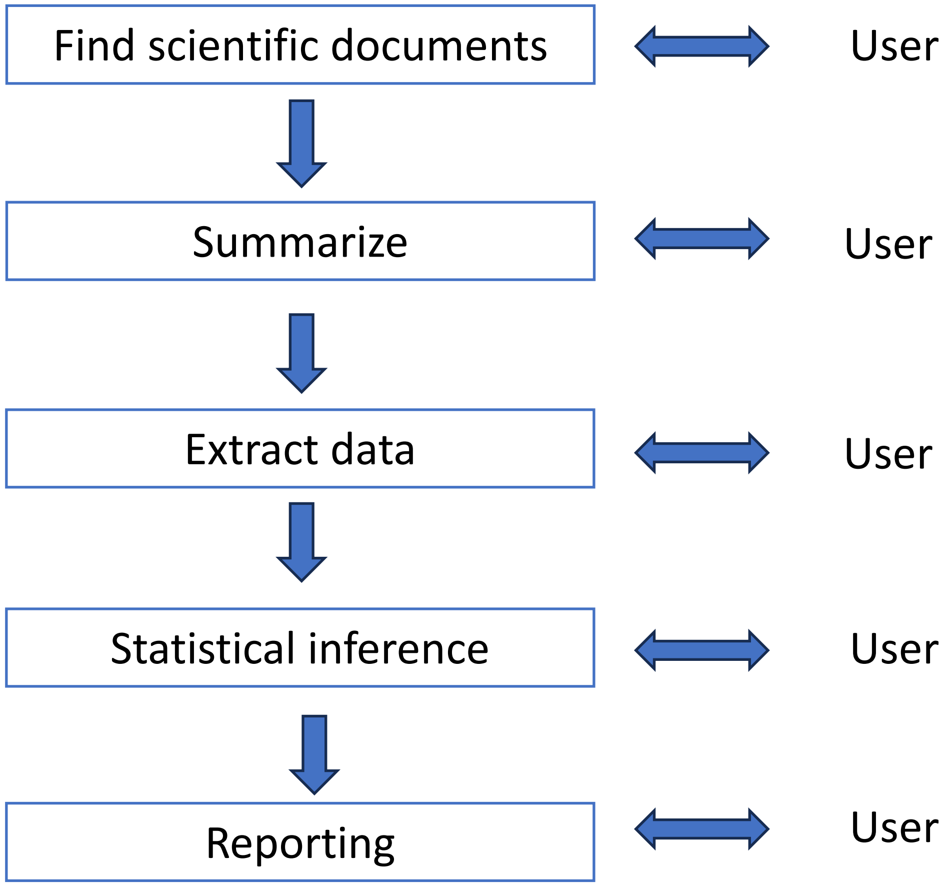
## Prompts that ask a Lot of Clarifying Questions



I'm not sure if I should put examples or conclusions first.

Would your audience benefit more from immediate clarity (conclusions) or from seeing practical applications first (examples)?

## LLM – enabled meta-analysis pipeline


Find scientific documents ⟷ User

Summarize ⟷ User

Extract data ⟷ User

Statistical inference ⟷ User

Reporting ⟷ User

# Part 2

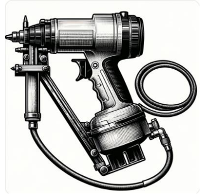
# LLMs for a Work-a-Day Power Planner

# Overarching Metaphor

## LLMs are a **nail gun** when we used to be **swinging hammers**

- A nail gun, by itself, does not make you a good carpenter
- A nail gun, by itself, does not build a house
- But if you are building a house, do you want a carpenter with a nail gun, or one with a hammer?

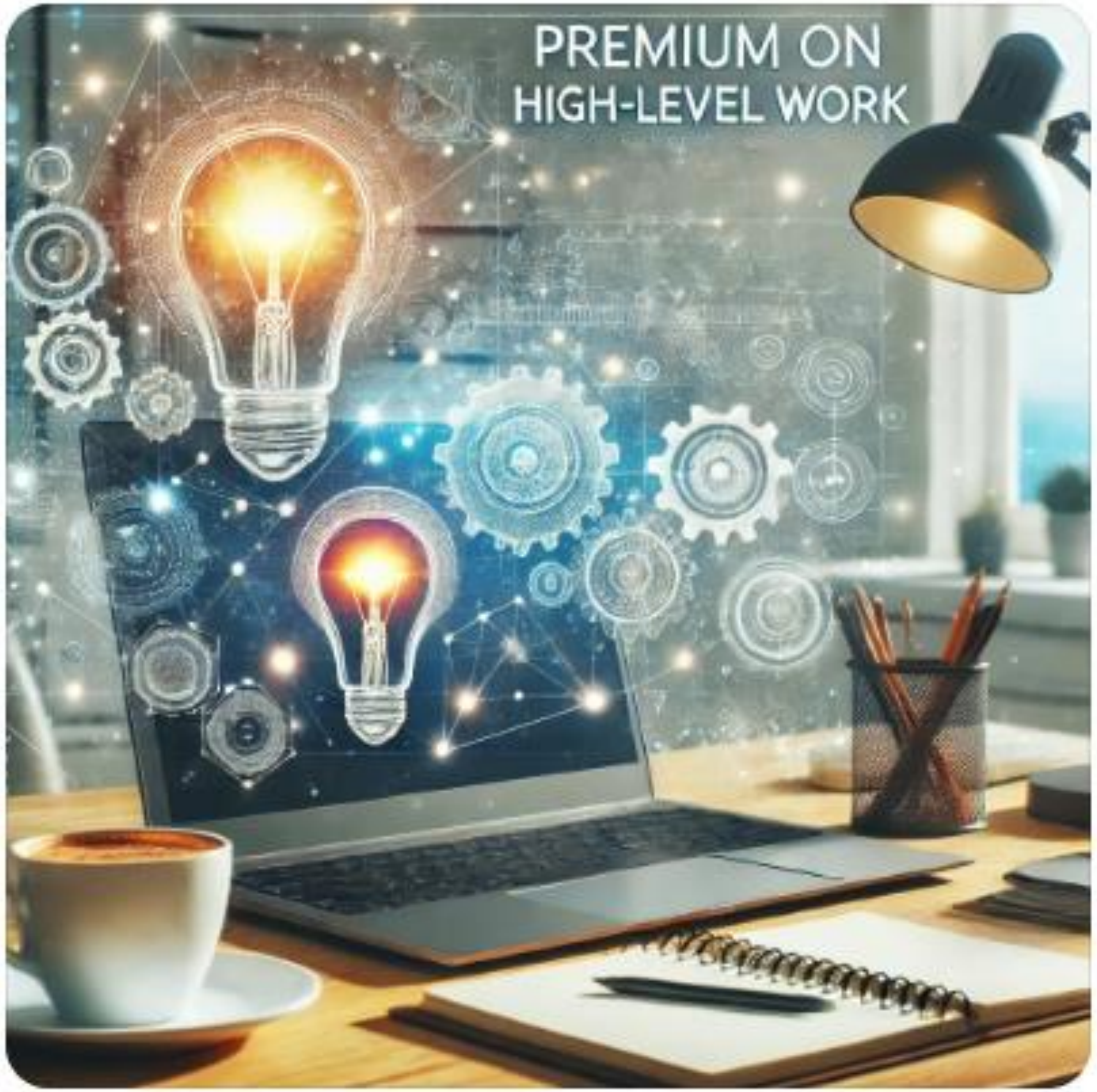
# Take Away #1



**LLMs put a premium on high level thinking**

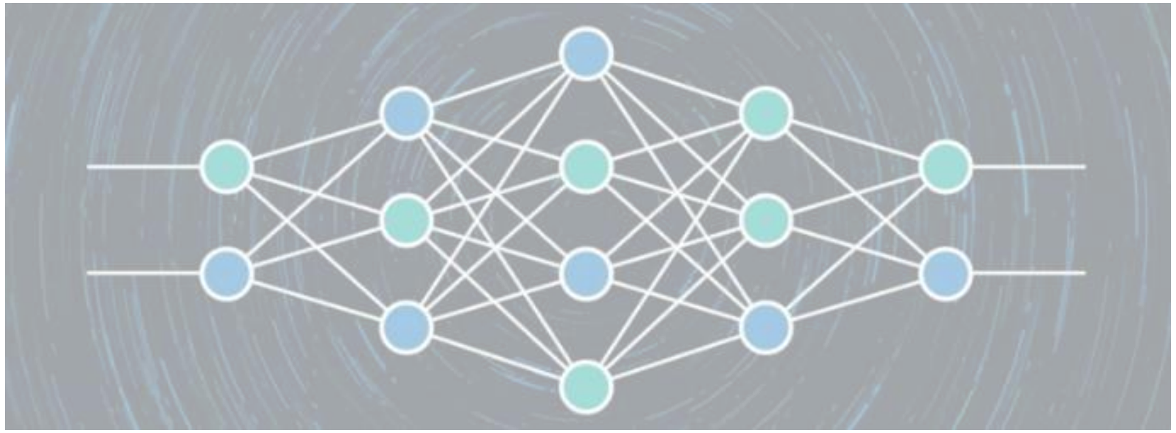
- The architect is important
- Be thoughtful about the overall project goals and structure
- Users need to keep up with the LLM

**Now is a good time to engage in meta-work!**



# What tasks are easier with LLMs?

- Tower of Babel tasks (translations in all directions)



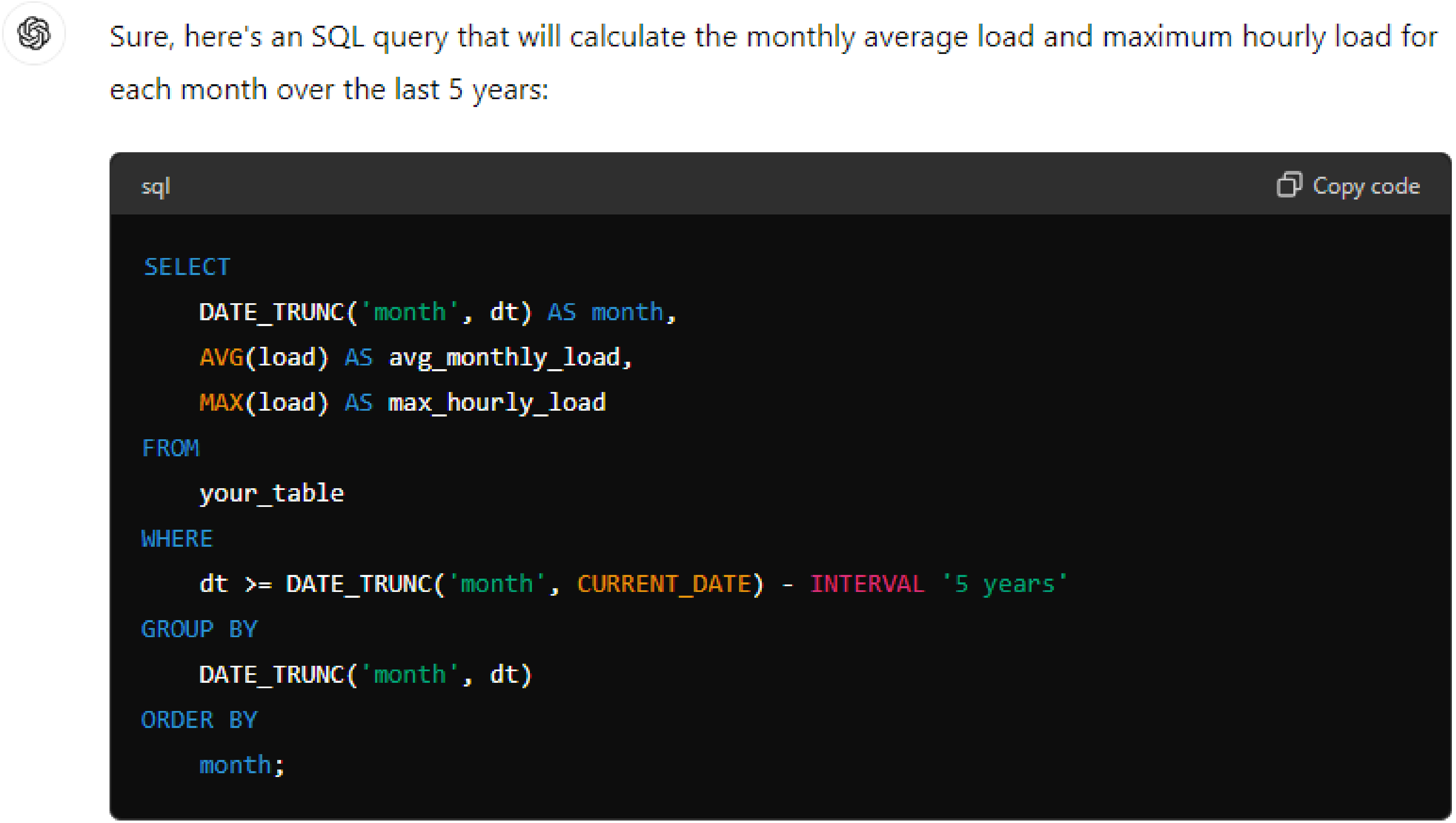
# Translation Example:
# Plain Language -> Computer Code

In the LLM prompt, type as much information as you can

LLM prompt

I have a **database** with a table with a datetime **column** called “dt” and a **column** of interest called “load.”  dt is hourly indexed.  **I would like** code to get the daily maximum loads for each month for the past 5 years.

# *Output*



Sure, here's an SQL query that will calculate the monthly average load and maximum hourly load for each month over the last 5 years:

```sql
SELECT
    DATE_TRUNC('month', dt) AS month,
    AVG(load) AS avg_monthly_load,
    MAX(load) AS max_hourly_load
FROM
    your_table
WHERE
    dt >= DATE_TRUNC('month', CURRENT_DATE) - INTERVAL '5 years'
GROUP BY
    DATE_TRUNC('month', dt)
ORDER BY
    month;
```

*You will probably get and error message when you try to run the code.*
*Iterate:*

# *Second Prompt*

Feed error messages back, along with your thoughts.

LLM prompt

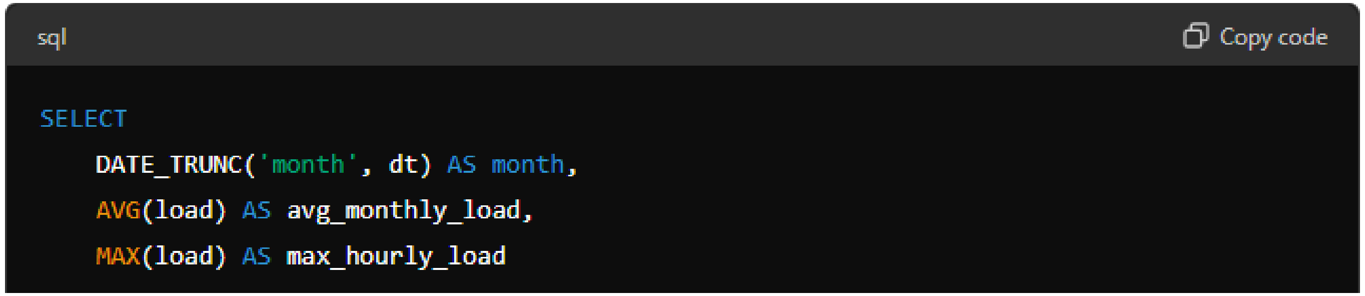
Oh, boy, I got an error message:

“ERROR……”

I think is has to do with the type of SQL . . .

# *Second Output*

The problem appears to be . . . here is the updated query. . .



```sql
SELECT
    DATE_TRUNC('month', dt) AS month,
    AVG(load) AS avg_monthly_load,
    MAX(load) AS max_hourly_load
```

. . .

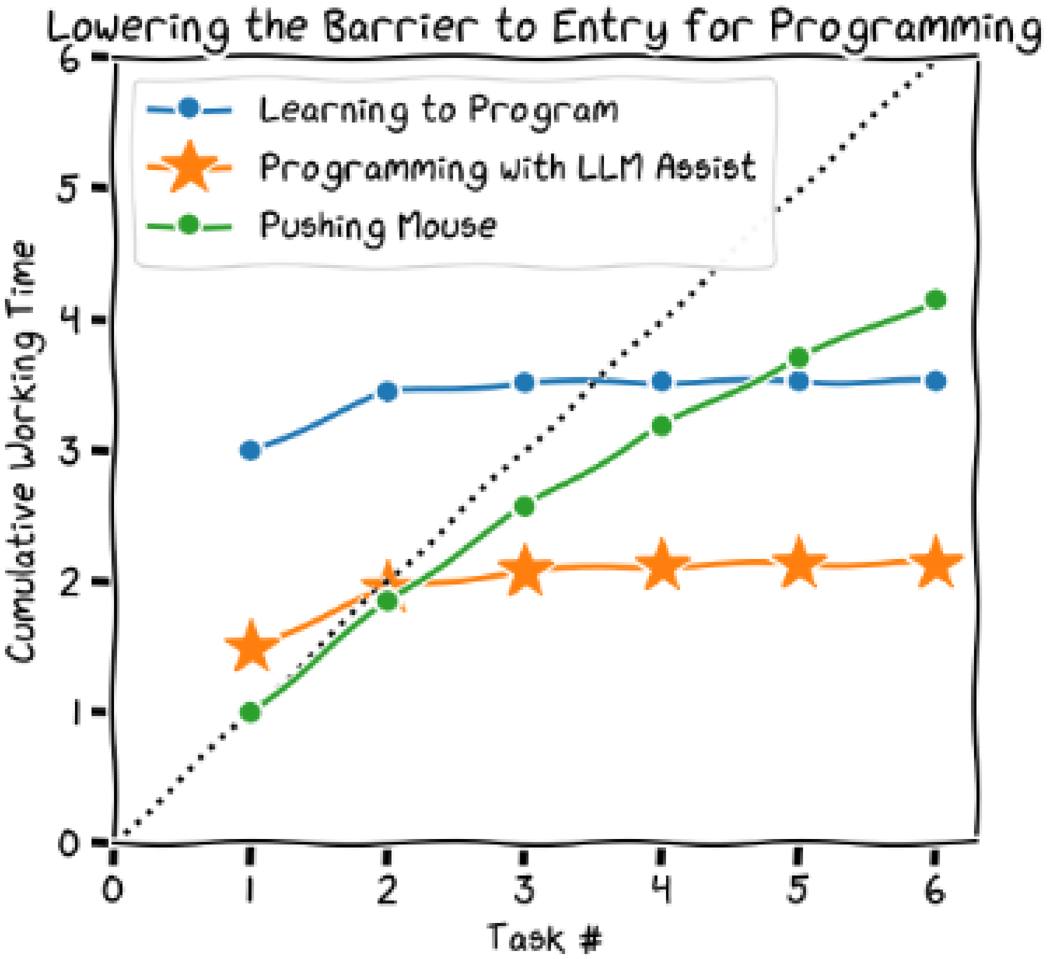
# What just happened

- If you are good at SQL, this saved you some time
- If you are bad at SQL, this saved you a ton of time
- If you can’t write SQL at all, you are starting to learn

# Take Away #2

**LLMs lower the barrier to entry for programming.**

Plain language -> programming language translation is a major event.



# Conceptually Simple, Technically Complex

**Goal: In modeling and data science work, use best practices from open-source software development**

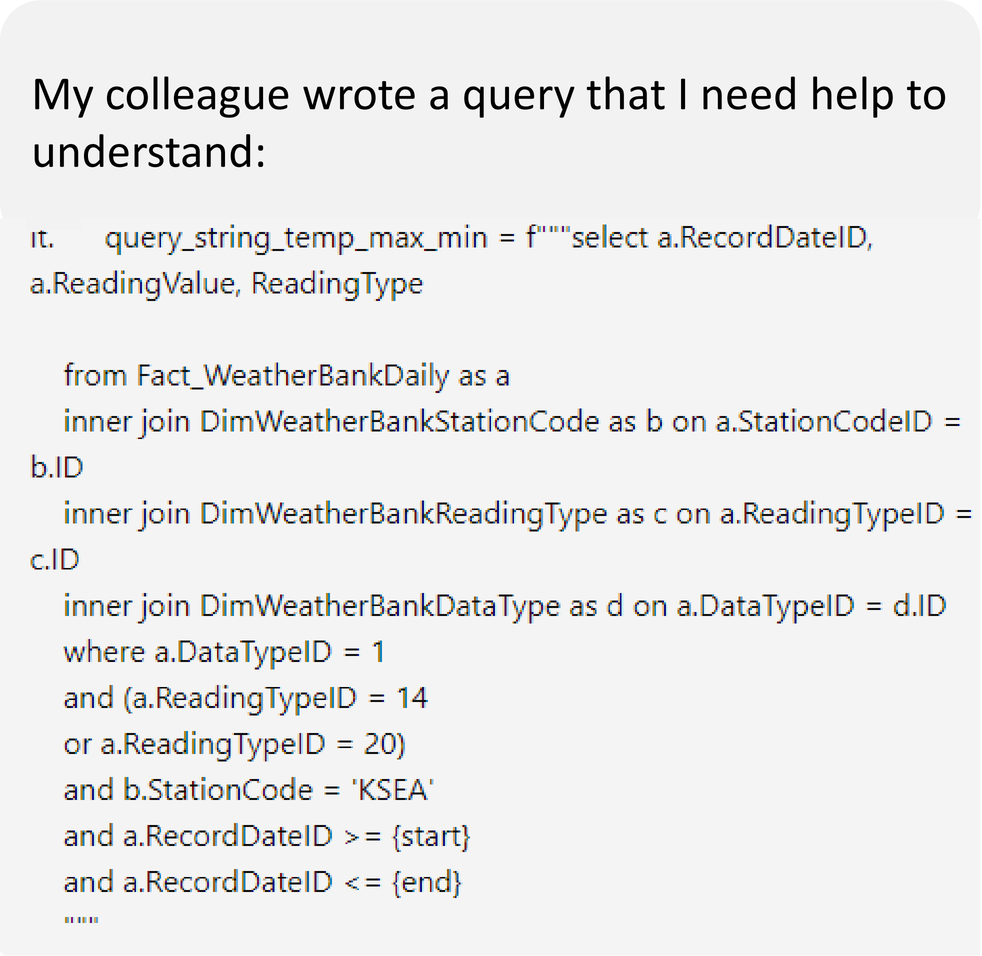
- There are a lot of things that are conceptually simple, but complex to implement
    - Git repository management (version control)
    - Database authorization and configuration
    - Code testing
    - Packaging and software dependency management
    - Documentation
    - Deployment

- **LLMs excel in these type of tasks**

# Translation Example:
# Computer Code -> Plain Language

Paste code into the chat and start asking questions

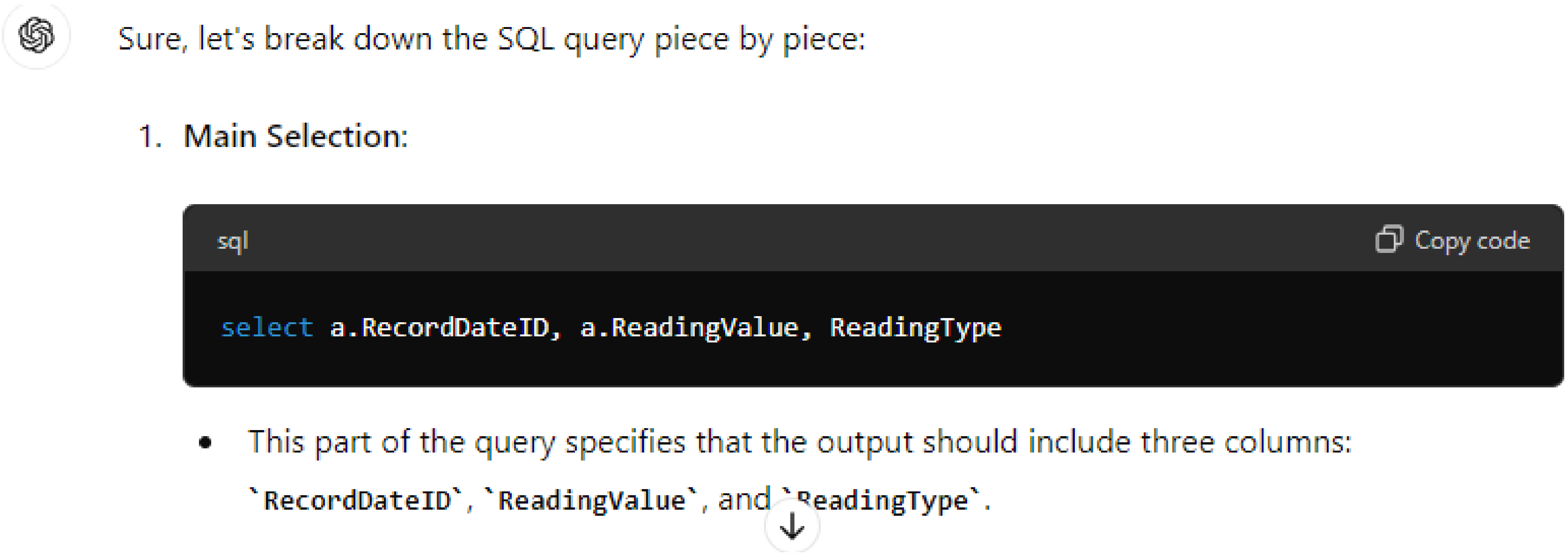
LLM prompt



My colleague wrote a query that I need help to understand:

```
ıt.     query_string_temp_max_min = f"""select a.RecordDateID,
a.ReadingValue, ReadingType

    from Fact_WeatherBankDaily as a
    inner join DimWeatherBankStationCode as b on a.StationCodeID =
b.ID
    inner join DimWeatherBankReadingType as c on a.ReadingTypeID =
c.ID
    inner join DimWeatherBankDataType as d on a.DataTypeID = d.ID
    where a.DataTypeID = 1
    and (a.ReadingTypeID = 14
    or a.ReadingTypeID = 20)
    and b.StationCode = 'KSEA'
    and a.RecordDateID >= {start}
    and a.RecordDateID <= {end}
    """
```

# *Output*



Sure, let's break down the SQL query piece by piece:

1. **Main Selection:**

```sql
select a.RecordDateID, a.ReadingValue, ReadingType
```

- This part of the query specifies that the output should include three columns: `RecordDateID`, `ReadingValue`, and `ReadingType`.

2. . . .

You will probably still not understand some parts and need to ask follow up questions.

# Translation Example:
# **Computer Code -> Computer Code from a Different Language**

## LLM prompts

In Excel, I would make a new column that adds column A with column B.  How would I do that in Python/ Pandas?

I have this Python code, can you translate it into R?

# Translation Example:
# Plain Language -> Data Visualization Code

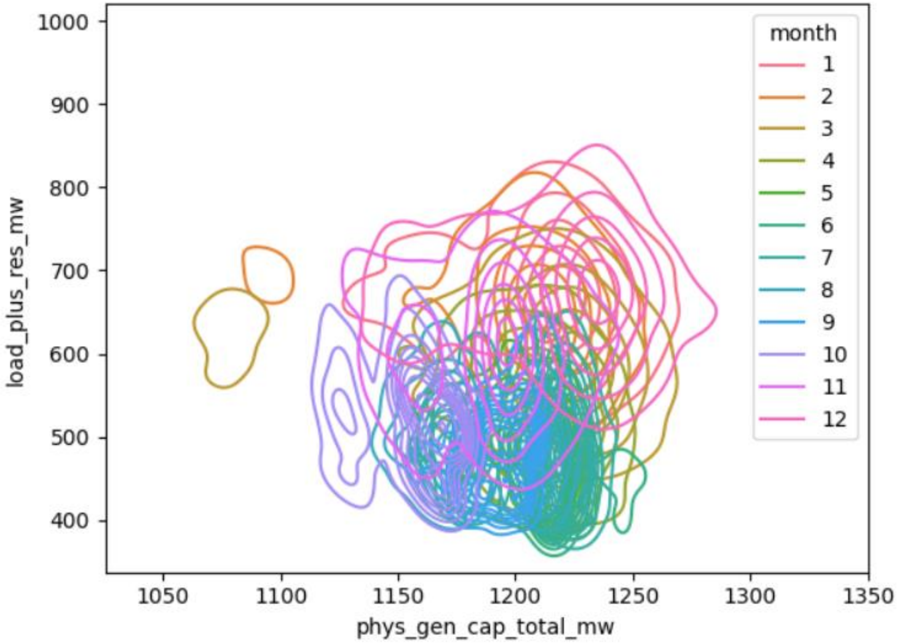
Good, insightful and replicable data visualization requires a lot of exploration and a lot of specific jargony computer code.

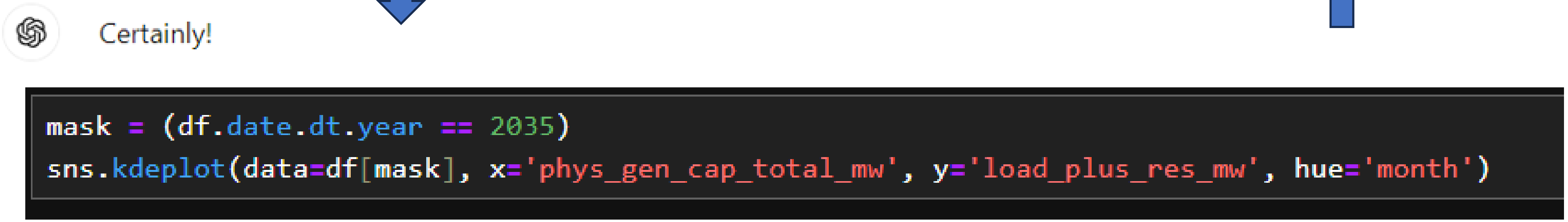
Let’s run through a back-and-forth workflow with an LLM.

# Plain Language -> Data Visualization Code

## LLM prompt

I have a dataframe with the following columns: […] I would like to make a two dimensional distribution plot with ‘columns x’, on the x and ‘column y’ on the y and the colors by month.  Only for the year 2035





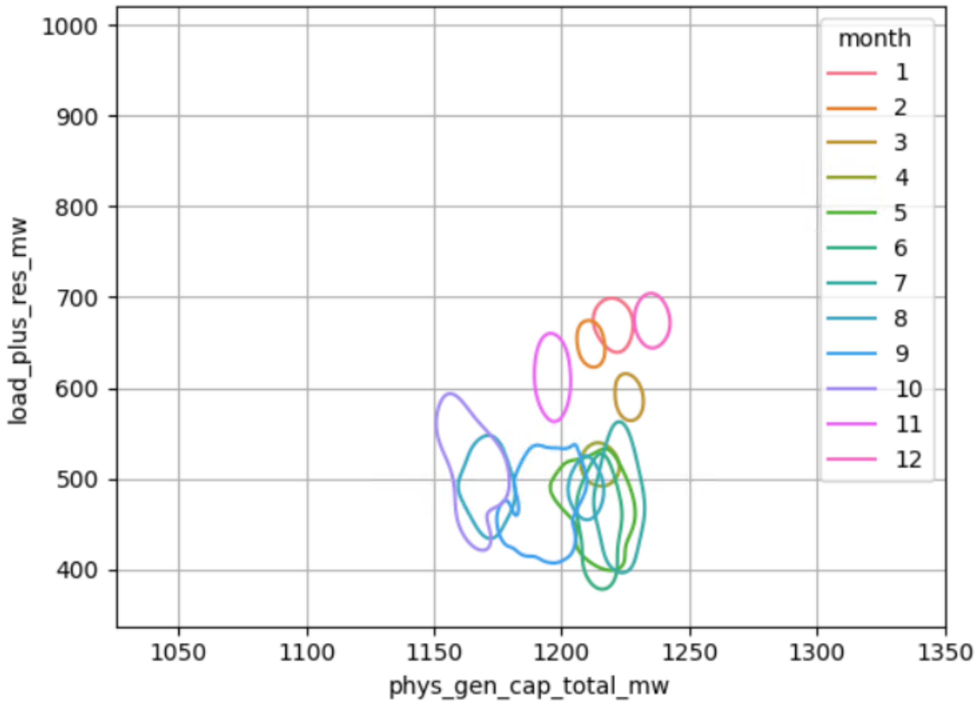
Certainly!

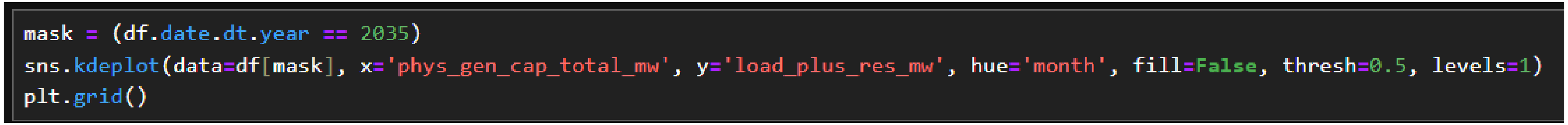
```
mask = (df.date.dt.year == 2035)
sns.kdeplot(data=df[mask], x='phys_gen_cap_total_mw', y='load_plus_res_mw', hue='month')
```

# Improve that Visualization

## LLM prompt

Whoa, that was a lot of lines. Could you make it just one contour line per month, maybe at the 50th percentile and add some grid lines?





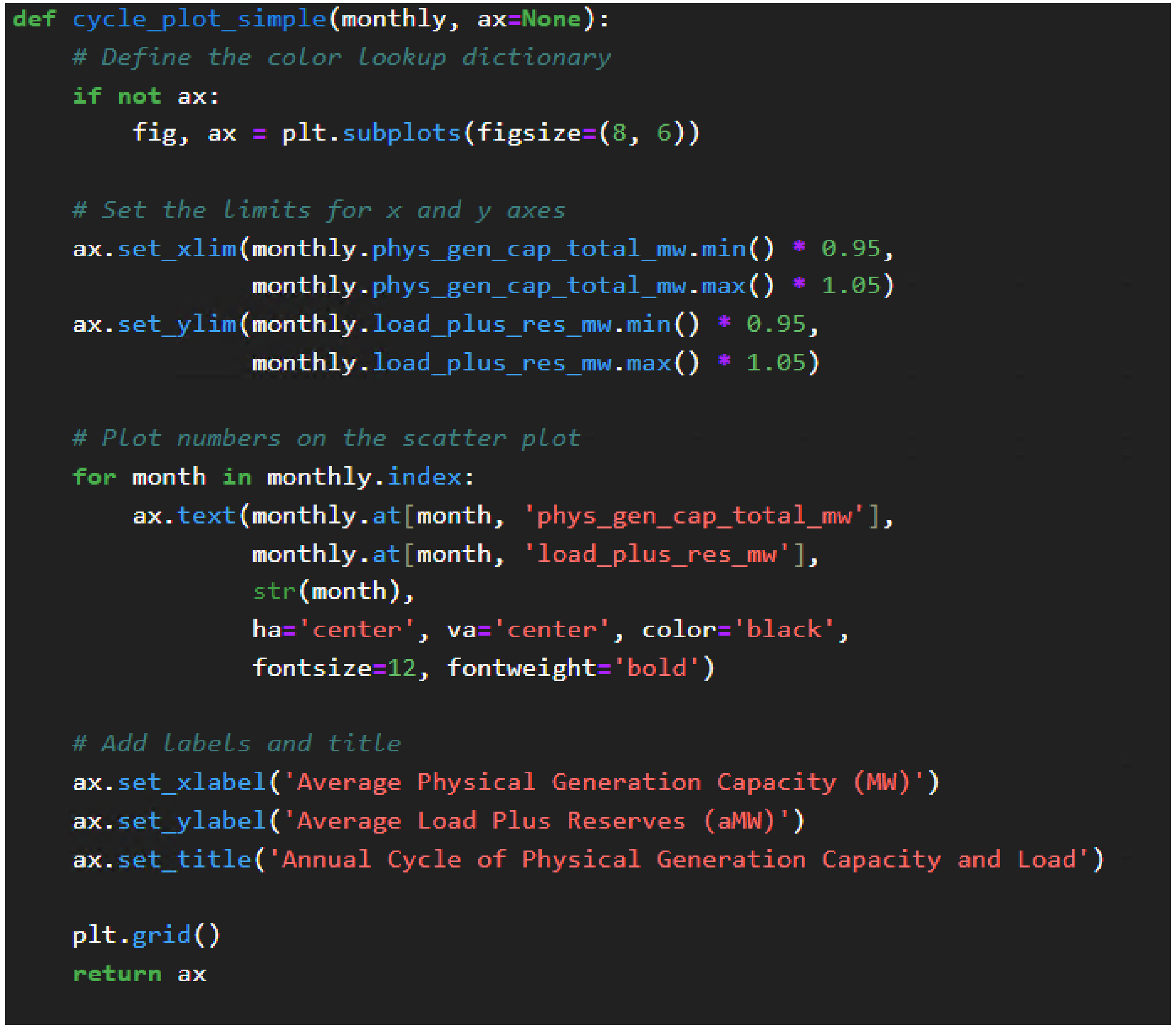
```
mask = (df.date.dt.year == 2035)
sns.kdeplot(data=df[mask], x='phys_gen_cap_total_mw', y='load_plus_res_mw', hue='month', fill=False, thresh=0.5, levels=1)
plt.grid()
```

# Improve that Visualization

## LLM prompt

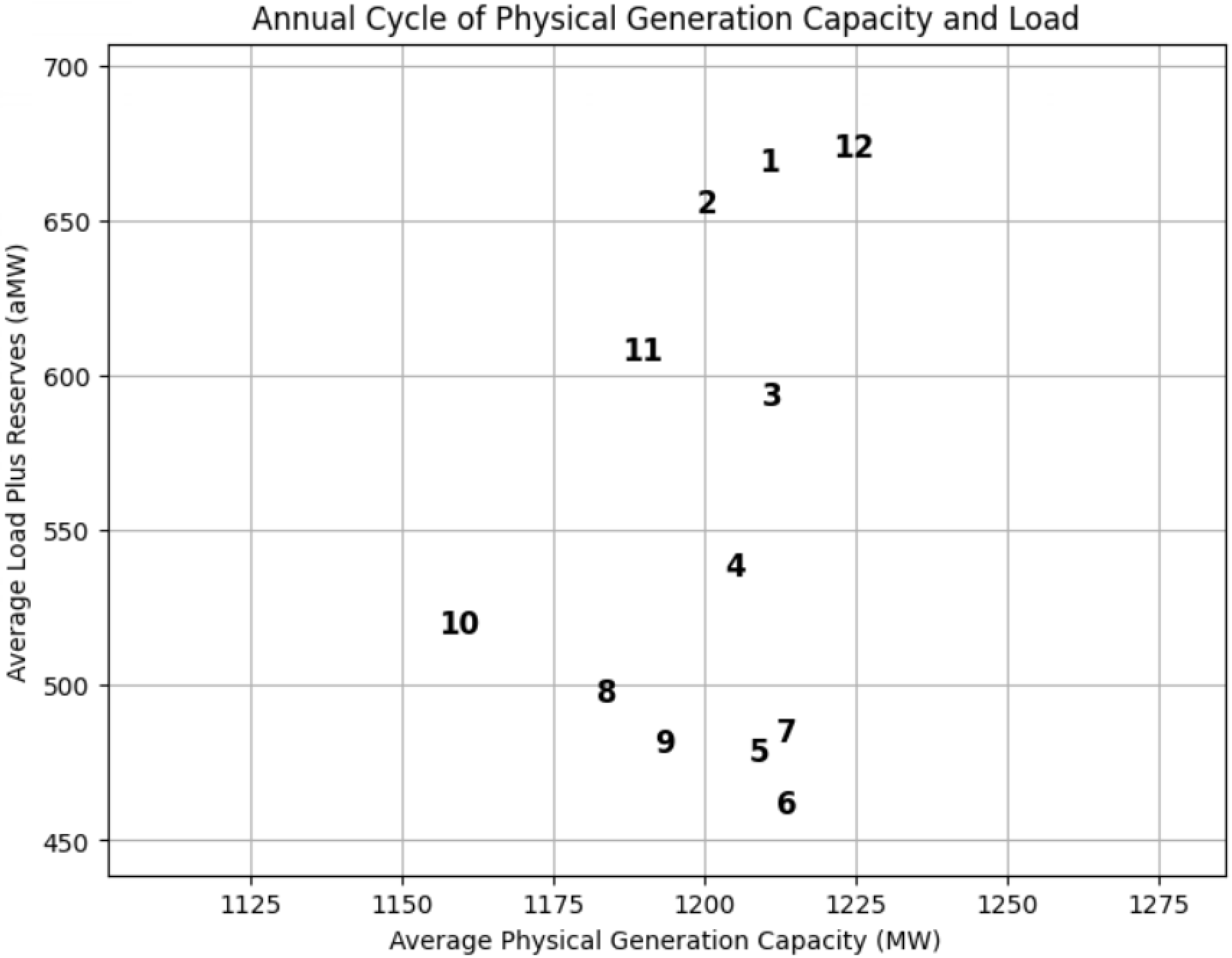
That is still hard to read, could we just put black numbers in the centroid for each month without the distribution plot with the same x and y axes?

Make sure to put the axis limits at + and – 5% from the min and max values, and label the axis and title the figure. And wrap the whole thing in a function.



```
def cycle_plot_simple(monthly, ax=None):
    # Define the color lookup dictionary
    if not ax:
        fig, ax = plt.subplots(figsize=(8, 6))

    # Set the limits for x and y axes
    ax.set_xlim(monthly.phys_gen_cap_total_mw.min() * 0.95,
                monthly.phys_gen_cap_total_mw.max() * 1.05)
    ax.set_ylim(monthly.load_plus_res_mw.min() * 0.95,
                monthly.load_plus_res_mw.max() * 1.05)

    # Plot numbers on the scatter plot
    for month in monthly.index:
        ax.text(monthly.at[month, 'phys_gen_cap_total_mw'],
                monthly.at[month, 'load_plus_res_mw'],
                str(month),
                ha='center', va='center', color='black',
                fontsize=12, fontweight='bold')

    # Add labels and title
    ax.set_xlabel('Average Physical Generation Capacity (MW)')
    ax.set_ylabel('Average Load Plus Reserves (aMW)')
    ax.set_title('Annual Cycle of Physical Generation Capacity and Load')

    plt.grid()
    return ax
```

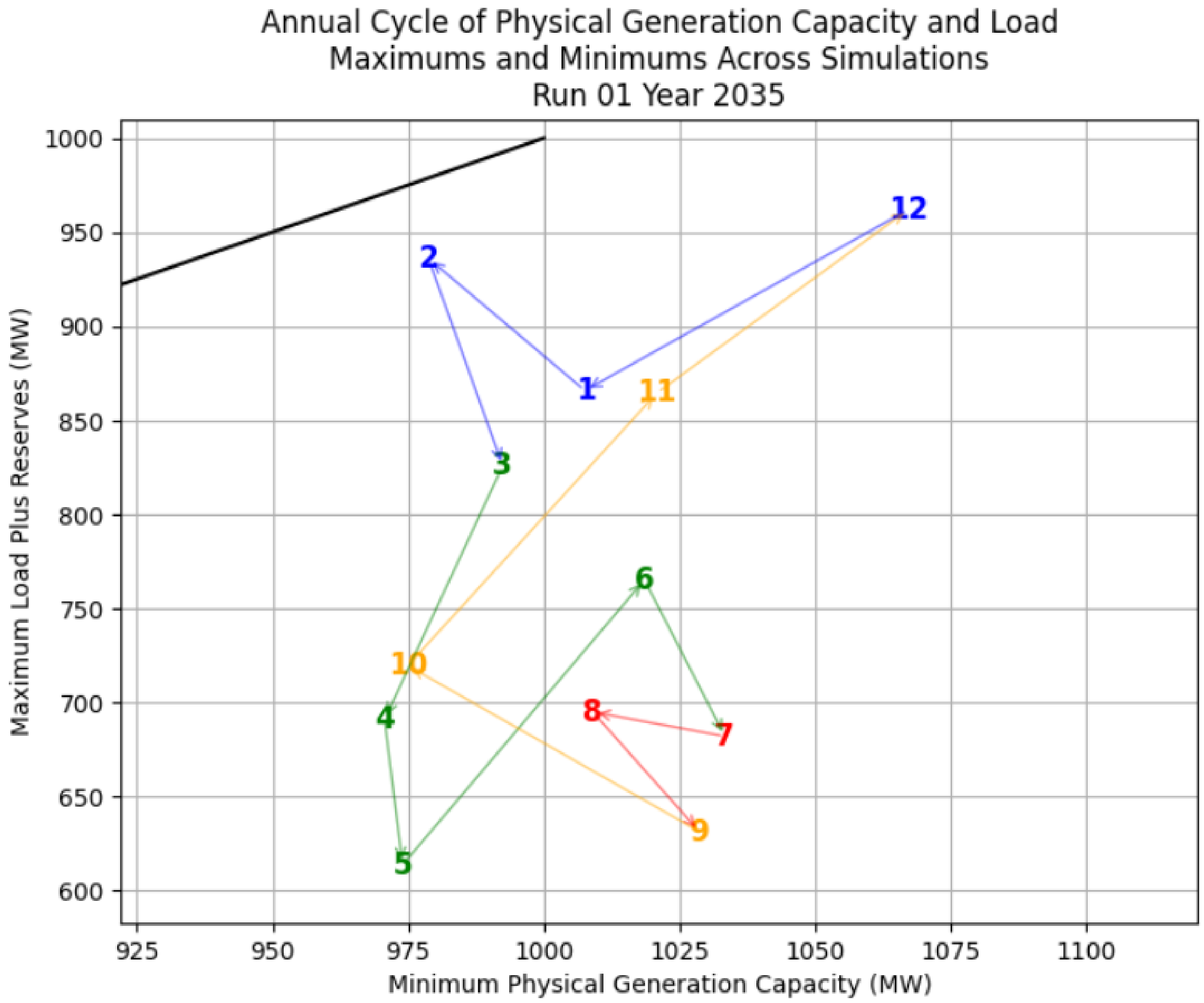
# Improve that Visualization



# Improve that Visualization

## LLM prompt

This is cool but still really hard to see. I need arrows between the month numbers on the graph and I also need the months numbers colored by season with winter blue, spring green, summer red, and fall orange. Oh, and I also need a one to one line on the plot to see when load might exceed capacity



Lots of Code

# Bonus Slide: Principles of Visual Display

1. Show Comparisons.
2. Provide Insight.
3. Present Multiple Dimensions/ Multivariate Data.
4. Integrate Evidence. Words, numbers, images, and diagrams should be fully integrated within a single presentation.
5. Provide Documentation.
6. Content matters most of all.

# Take Away #3

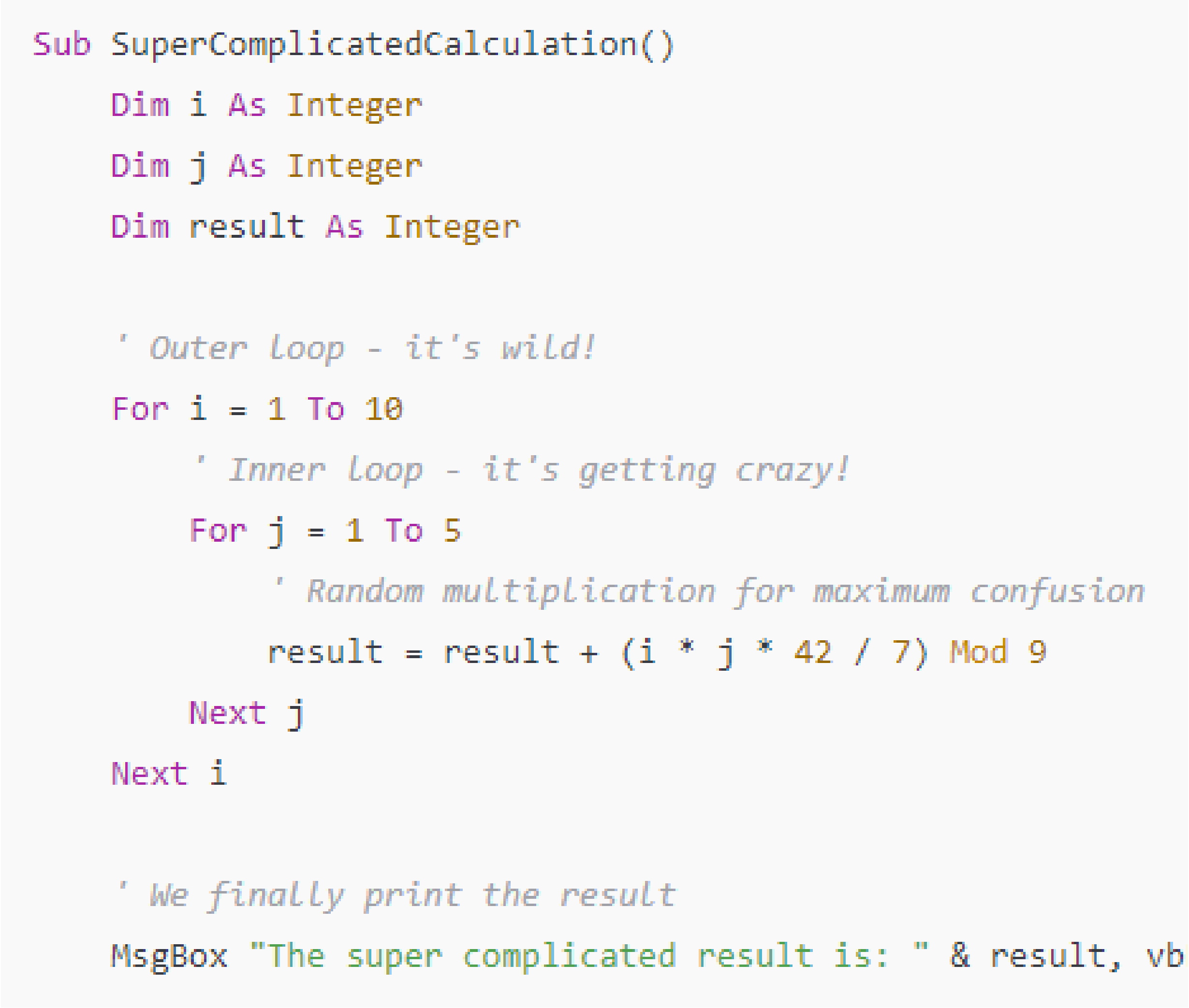
**LLMs are making it much easier to switch programming languages.**

> "Are you an R shop, a Python shop, a Julia shop, a bunch of Excel hacks?"
>
> "Yes"

# Take Away #4

**LLMs make it easier to write in difficult programming languages.**

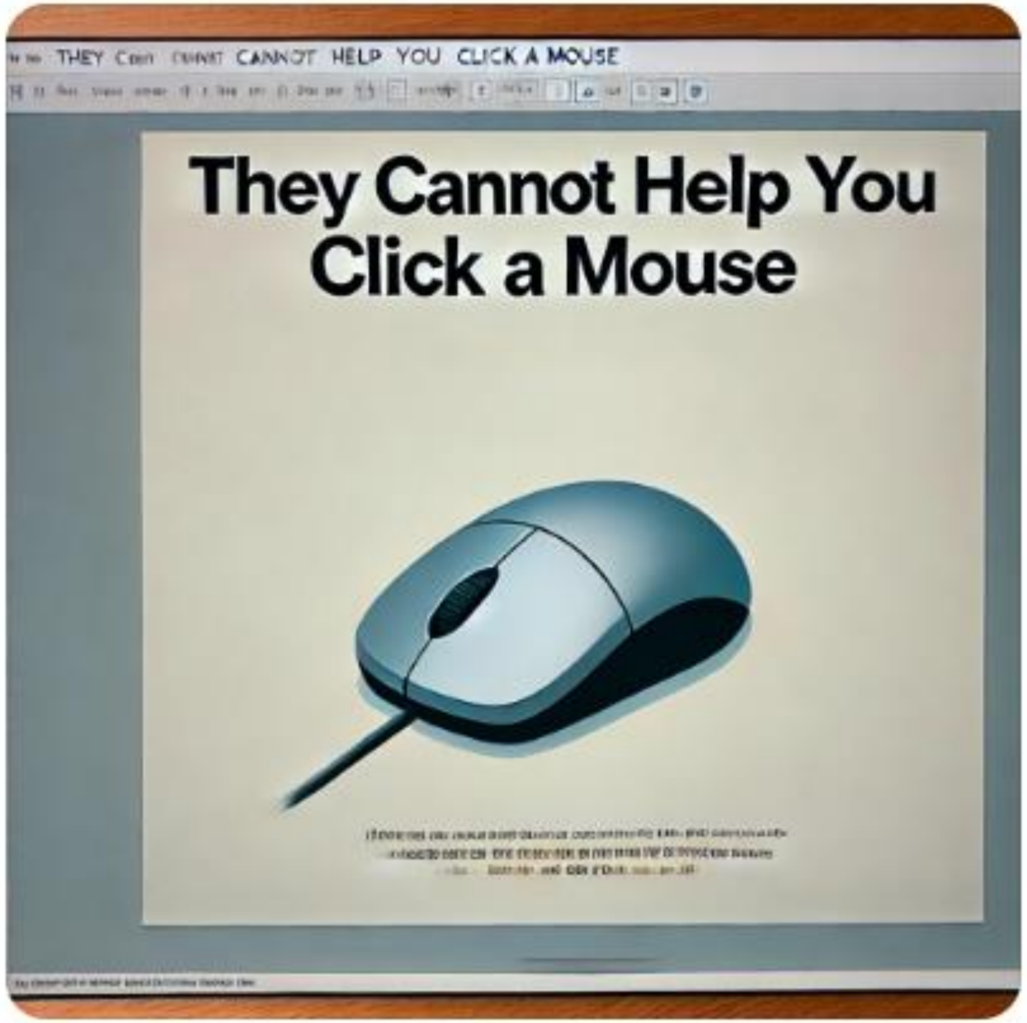
> "I.T. will only let us use Visual Basic. . . "
>
> "No problem"



```
Sub SuperComplicatedCalculation()
    Dim i As Integer
    Dim j As Integer
    Dim result As Integer

    ' Outer loop - it's wild!
    For i = 1 To 10
        ' Inner loop - it's getting crazy!
        For j = 1 To 5
            ' Random multiplication for maximum confusion
            result = result + (i * j * 42 / 7) Mod 9
        Next j
    Next i

    ' We finally print the result
    MsgBox "The super complicated result is: " & result, vb
```

# Take Away #5

LLMs are ***language*** models.


THEY CANNOT HELP YOU CLICK A MOUSE
They Cannot Help You Click a Mouse

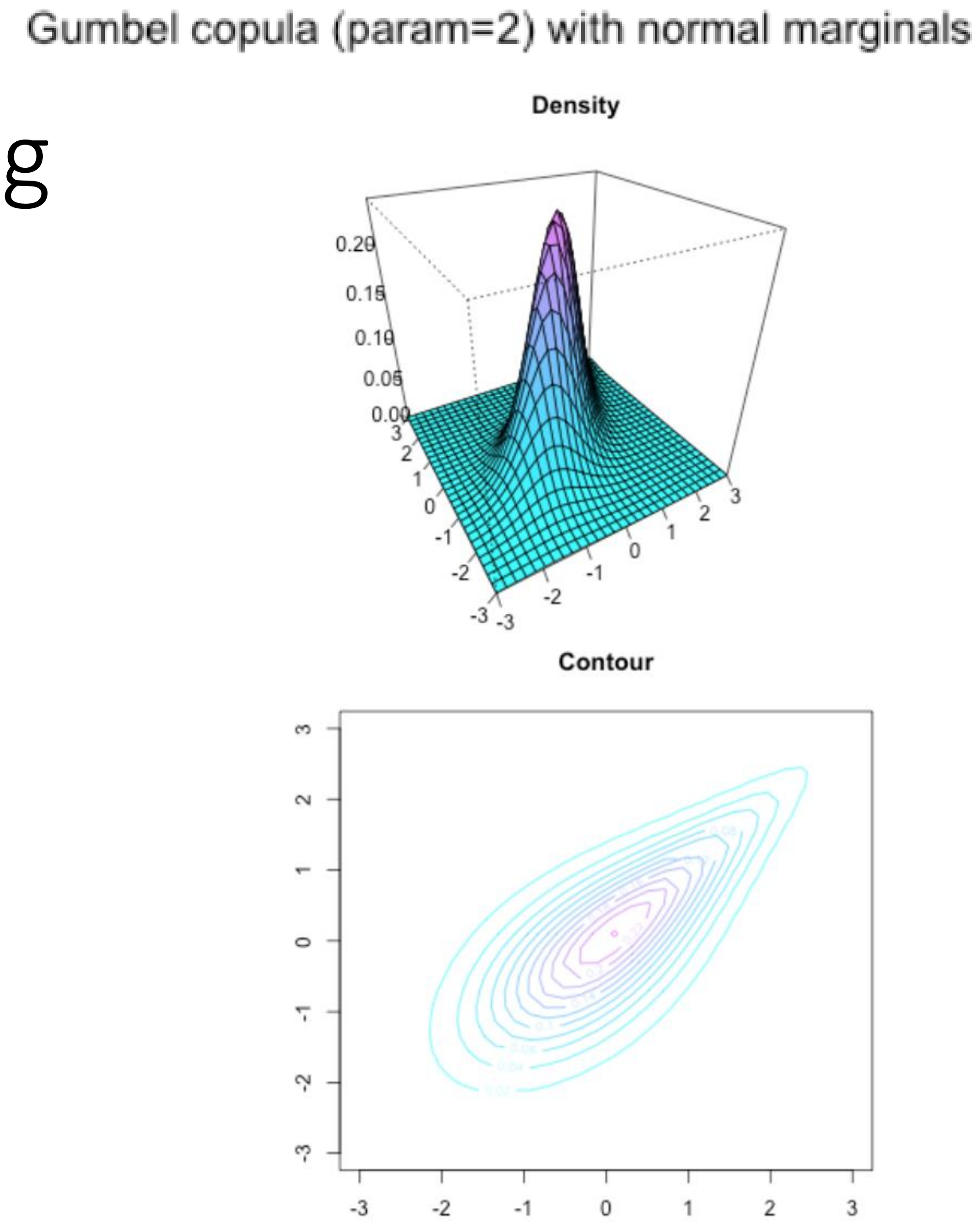
# How Can LLMs Help the Mousers?

- ??? I don’t know ???
- Lower barrier to entry for programming?
- Wait around for power-charged LLM-augmented graphical programs to improve?

# Part 3. LLLs and LLMs
# Life-long Learners and Large Language Models

# LLMs as an Aid to Learning

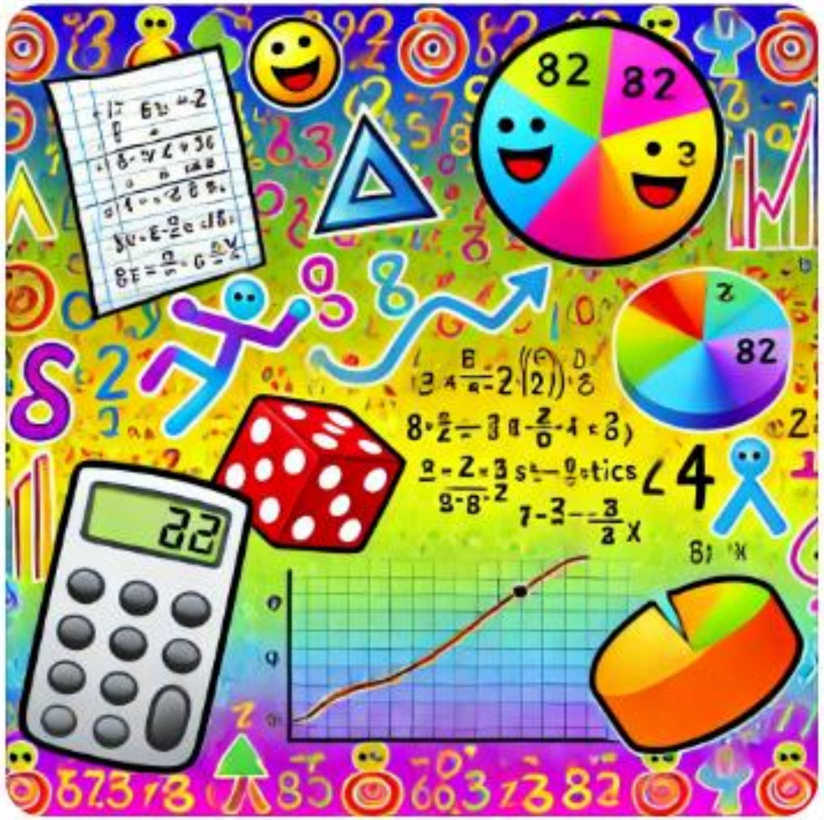
- Example: what the heck is a copula?
  - Ask AI to: write out the mathematical form
  - Make that into a Python function to play with
  - Generate example data
  - Visualize that example data
  - Repeat



# LLMs as an Aid to Learning

## Current Sub-Projects

- Python design patterns
- Hydropower management
- Machine learning
- Linear optimization
- Reinforcement learning
- Climate projections
- Production cost modeling



# LLM Fails and Successes

**Fails**

- Thinking the LLM is better than it is – especially a precise understanding context
- Debugging failure loops
    - Especially if I don't have the background knowledge
- Canned, banal, vapid and verbose writing style

**Successes**

- General coding, especially common applications
- Organizing information and creating outlines
- Summarizing

# Humanist Perspective on Artificial Intelligence

**AI is about. . .**

- AI is about helping people and organizations
- AI is about automating the boring stuff
- AI is about doing the things you want to do

**AI is NOT about. . .**

- AI is NOT about selling stuff
- AI is NOT about spamming and scamming people
- AI is NOT about replacing humans

*AI is putting a premium on* ***human*** *communication and* ***human*** *intelligence*

# Call to Action

Be the architect

Keep your nail gun handy

Don’t be bored