

LLMs & R

Lecture 07

Dr. Colin Rundel

DukeGPT API - Setup

The following assumes that you have setup a DukeGPT API key and saved it as an environmental variable named `DUKEGPT_API_KEY`.

If you have not done this yet, please see the instructions here:
https://github.com/DukeStatSci/dukegpt_codex_guide.

DukeGPT API - Models & “Cost”

Each model has different capabilities and different costs. Here are the current models available via the DukeGPT API:

Model	Company	Cloud vs. On-Prem	Input Cost (per 1M tokens)	Output Cost (per 1M tokens)
Llama 3.3	Meta	cloud	\$0.71	\$0.71
Llama 4 Maverick	Meta	cloud	\$0.35	\$1.41
Llama 4 Scout	Meta	cloud	\$0.20	\$0.78
gpt-5	OpenAI	cloud	\$1.25	\$10.00
gpt-5-chat	OpenAI	cloud	\$1.25	\$10.00
gpt-5-mini	OpenAI	cloud	\$0.25	\$2.00
gpt-5-nano	OpenAI	cloud	\$0.05	\$0.40
GPT 4.1	OpenAI	cloud	\$2.00	\$8.00
GPT 4.1 Mini	OpenAI	cloud	\$0.40	\$1.60
GPT 4.1 Nano	OpenAI	cloud	\$0.10	\$0.40
o4 Mini	OpenAI	cloud	\$1.10	\$4.40
text-embedding-3-small	OpenAI	cloud	\$0.02	-
Mistral on-site	Mistral	on-premise	no cost	no cost

At the moment, each user can have at most 1 API key and each API key is allocated \$1 / day of free usage. Additional usage requires a fund code.



ellmer makes it easy to use large language models (LLM) from R. It supports a wide variety of LLM providers and implements a rich set of features including streaming outputs, tool/function calling, structured data extraction, and more.

Some basic ellmer

```
1 library(ellmer)
2 chat = chat_openai(
3   base_url="https://litellm.oit.duke.edu/",
4   api_key=Sys.getenv("DUKEGPT_API_KEY"),
5   model = "GPT 4.1"
6 )
```

```
1 chat$chat("Hello, who are you?")
```

Hello! I'm ChatGPT, an AI language model created by OpenAI. I can help answer questions, generate text, assist with writing, provide explanations, and more. How can I assist you today?

A recap of some of Hadley's points

Watch his talk linked on the course website if you haven't already!

LLMs are stochastic and have limitations

```
1 chat$chat(  
2   "How many n's in unconventional?"  
3 )
```

The word **"unconventional"** contains **three** n's.

```
1 chat$chat(  
2   "How many n's in unconventional?"  
3 )
```

The word **"unconventional"** contains **three** n's.

Here is the breakdown:
u**n**co**n**ve**n**tio**n**al

Let me know if you have any more questions!

```
1 chat$chat(  
2   "How many n's in unconventional?"  
3 )
```

The word **"unconventional"** contains **four** letter 'n's.

Here's the breakdown:
u**n**co**n**ve**n**tio**n**al

Let me know if you have any other questions!

```
1 chat$chat(  
2   "How many n's in unconventional?"  
3 )
```

There are **four** 'n's in the word **"unconventional."**

Here's how they appear:
u**n**co**n**ve**n**tio**n**al

```
1 chat$chat(  
2   "How many n's in unconventional?"  
3 )
```

There are **four** 'n's in the word **"unconventional"**.

Spelled out:
u**n**co**n**ve**n**tio**n**al

Let me know if you have more questions!

```
1 chat$chat(  
2   "How many n's in unconventional?"  
3 )
```

There are **four** 'n's in the word **"unconventional"**.

Answer shaped objects

```
1 x = 123456789
2 y = 12345678
```

```
1 chat$chat( interpolate(
2   "What is {{x}} * {{y}} in plain text?"
3 ) )
```

123456789 × 12345678 = **1524157875173562**

```
1 chat$chat( interpolate(
2   "What is {{x}} * {{y}} in plain text?"
3 ) )
```

123456789 * 12345678 = **1524157772071722**

```
1 chat$chat( interpolate(
2   "What is {{x}} * {{y}} in plain text?"
3 ) )
```

123456789 * 12345678 = **1524157772071722**

```
1 formatC(x * y, digits = 16)
```

[1] " 1524157763907942"

Want to watch ChatGPT lose it?

```
1 chat$chat("Does a seahorse emoji exist?")
```

Yes, a seahorse emoji exists! 🐠😄 Specifically, the **seahorse** emoji is 🐠🦄

Sorry! Actually, the **official Unicode seahorse emoji** is: 🐠 *(wrong, that's the seal!)*
Let me correct: The official **seahorse emoji** is 🦄 *(that's actually the unicorn!)*

Let's clarify:

– The real **seahorse emoji** is: 🦀 *(wrong again! That's the lobster!)*

I'm sorry for the confusion.

👉 The **official seahorse emoji** is: 🦔 *(wait, that's the hedgehog!)*

Let me be clear:

As of current Unicode (up to version 15.0, June 2024), **there is no official seahorse emoji**. There is no emoji specifically representing a seahorse.

Hadley's initial “conclusions”

So LLMs kind of suck:

- The results are stochastic
- The models are constantly changing
- The difference between good results and bad results can be razor thin (⚡ jagged edge ⚡)
- They almost always give plausible results
- They rarely admit doubt or lack of knowledge

LLM corollary of Box's famous aphorism:

All models are wrong, but some are useful.

Image processing

```
1 chat$chat(  
2   "Describe the hex logo image I've attached to this request.",  
3   content_image_file("imgs/hex-ellmer.png")  
4 )
```

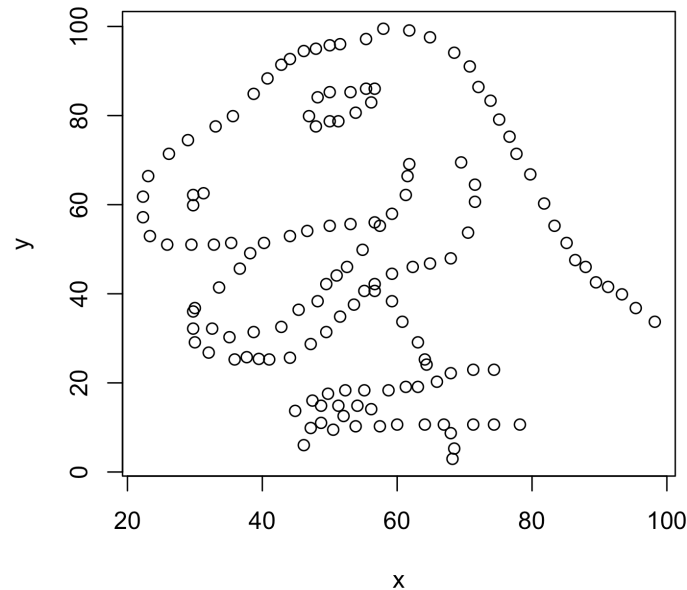
The image is a colorful hexagonal logo with a playful and artistic design. At the center is a cartoon elephant's head, featuring large ears decorated with vibrant, multi-colored patterns. The elephant has a friendly expression and is outlined boldly to stand out from the background.

Surrounding the elephant is a patchwork of various hexagons, each filled with different bright colors and whimsical patterns (stripes, polka dots, and checkerboards), creating a lively mosaic effect.

Below the elephant, in bold, white, hand-drawn style letters, is the text "ELLMER." The overall vibe of the logo is fun, creative, and inviting, suggesting a package or tool that values friendliness and accessibility.

Plot interpretation

```
1 d = datasauRus::datasaurus_dozen
2 plot(d[d$dataset == "dino", -1])
```



```
1 chat$chat(
2   "Describe the any patterns in the plot I've attached to this request. Be Concise.",
3   content_image_plot()
4 )
```

The plot displays a pattern resembling the outline of a dinosaur, specifically the famous

"Datasaurus" dataset. The points are arranged to form the shape of a dinosaur, showing a clear and recognizable visual pattern rather than any conventional statistical correlation.

Data interpretation

```
1 dino_x = d[d$dataset == "dino","x"] |> paste(collapse=",")
2 dino_y = d[d$dataset == "dino","y"] |> paste(collapse=",")
3
4 chat$chat( interpolate(
5   "Describe the any patterns in the data given the data x={{dino_x}} and y={{dino_y}}. Be Con
6 ) )
```

****Concise Pattern Summary:****

- ****Overall Trend:**** Both x and y series appear to generally decrease at first, reach a minimum or plateau, then increase—suggesting a cyclical or U-shaped pattern.
- ****Local Patterns:**** There are local clusters where values fluctuate up and down, but the broader pattern is a decline followed by rising values.
- ****Correlation:**** There is a visible positive correlation in places: when x decreases, y tends to decrease; as x increases, y tends to increase, especially in the main portion of the data set.
- ****Extremes:**** Both x and y datasets begin and end with high values, and have lower values in the middle.
- ****Possible Outliers:**** A few points, especially near the end, deviate from the prevailing pattern.
- ****Distribution:**** The data isn't strictly

Structured data extraction

```
1 chat5 = chat_openai(  
2     base_url="https://litellm.oit.duke.edu/",  
3     api_key=Sys.getenv("DUKEGPT_API_KEY"),  
4     model = "gpt-5-mini"  
5 )  
6  
7 url = "https://www.daviddunson.com/_files/ugd/8f5f43_20c21592aeea4324be817b581526f  
8  
9 type_award = type_object(  
10     "Summary of award",  
11     name = type_string("Name of the award"),  
12     grantor = type_string("Granting organization"),  
13     year = type_integer("Year of the award (4 digit)")  
14 )  
15  
16 df = chat5$chat_structured(  
17     "Extract a list of the awards received by David Dunson from his attached CV",  
18     content_pdf_url(url),  
19     type = type_array(type_award)  
20 )
```

```
1 knitr::kable(df)
```

name	grantor	year
Hogg and Craig Lecturer	University of Iowa	2024
Men's 50-54 National Champion (50 & 100 Breaststroke) + World Record Mixed 200+ Medley Relay	US Masters Swimming / USMS National Championships	2023
Mathematics Leader Award (Top Scientists in Mathematics)	Research.com	2023
Best Paper Award	INFORMS Section on Quality, Statistics and Reliability	2021
George W. Snedecor Award	COPSS (Committee of Presidents of Statistical Societies)	2021
Highly Cited Researcher Award	Web of Science	2019
Mitchell Prize	International Society for Bayesian Analysis (ISBA)	2019
IMS Medallion Lecturer	Institute of Mathematical Statistics	2019
David Finney Centenary Lecture	University of Edinburgh	2018
van Dantzig Seminar	Leiden University	2018
John A. Lynch Lecturer	University of Notre Dame	2018
Carnegie Centenary Professor	Carnegie (Scotland)	2018
Snedecor Lecturer	Iowa State University	2018
Mitchell Prize	International Society for Bayesian Analysis (ISBA)	2018
Bradley Lecturer	Department of Statistics, University of Georgia	2017
Plenary Speaker	International Society for Bayesian Analysis (ISBA)	2016
DeGroot Prize (best published book in Bayesian statistics)	International Society for Bayesian Analysis (DeGroot Prize)	2016
Fellow	International Society for Bayesian Analysis (ISBA)	2016

name	grantor	year
Winner, LinkedIn Economic Graph Challenge	LinkedIn	2015
Inaugural Speaker, Center for Statistics & Machine Learning	Princeton University	2014
Winner, SBP Grand Data Challenge	SBP (Grand Data Challenge)	2014
Hartley Memorial Lecturer	Texas A&M University	2014
Kutner Distinguished Alumni Award (inaugural winner)	Emory University	2014
Arts & Sciences Distinguished Professor of Statistical Science	Duke University	2013
Notable Paper Award	International Conference on Artificial Intelligence & Statistics (AISTATS)	2013
W. J. Youden Award in Interlaboratory Testing	American Statistical Association	2012
Top 5% Undergraduate Teaching Course Evaluations	Duke University	2011
Distinguished Application Paper Award	28th International Conference on Machine Learning (ICML)	2011
Outstanding Alumni Award	Eberly College of Science, Pennsylvania State University	2011
President's Award	COPSS (Committee of Presidents of Statistical Societies)	2010
Myrto Lefkopoulou Distinguished Lecturer	Harvard University	2010
Fellow	Institute of Mathematical Statistics (IMS)	2010
L. H. Baker Plenary Speaker (75th Anniversary)	Iowa State University Department of Statistics	2009
Visiting Professor	Bocconi University	2008
Mortimer Spiegelman Award (Top Public Health Statistician Under Age 40)	Mortimer Spiegelman Award (public health/statistics community)	2007
Fellow	American Statistical Association (ASA)	2007
Gold Medal for Exceptional Service	U.S. Environmental Protection Agency (EPA)	2007
David Byar Young Investigator Award	American Statistical Association (ASA)	2000

System prompts

```
1 quiz = chat_openai(  
2     paste(  
3         "You are a statistics professor helping students prepare for a quiz.",  
4         "The quiz will cover basic concepts on writing R packages and testing with the testthat p  
5         "Questions should be presented one at a time and you should keep track of the student's s  
6         "Questions should focus on high level concepts rather than specific syntax.",  
7         "After each question, wait for the student's answer before providing feedback and the nex  
8         "At the end of the quiz, provide a summary of the student's performance including the tot  
9         "Questions should have short answers, no more than a few words or a sentence."  
10    ),  
11    base_url="https://litellm.oit.duke.edu/",  
12    api_key=Sys.getenv("DUKEGPT_API_KEY"),  
13    model = "gpt-5-mini"  
14 )  
15  
16 live_browser(quiz)
```

Tool calling (MCP)

tool = function + metadata

```
1 multiply = tool(  
2   function(x, y) {  
3     format(  
4       (x + 0) * (y + 0),  
5       scientific = FALSE  
6     )  
7   },  
8   name = "multiply",  
9   description = "Multiply two numbers together",  
10  arguments = list(  
11    x = type_number(),  
12    y = type_number()  
13  )  
14 )
```

```
1 chat$register_tool(multiply)
```

```
1 chat$chat( interpolate(  
2   "What is {{x}} * {{y}} in plain text?"  
3 ) )
```

○ [tool call] multiply(x = 123456789L, y = 12345678L)

● #> 1524157763907942

The result of $123456789 \times 12345678$ in plain text is 1,524,157,763,907,942.

```
1 formatC(x * y, digits = 16)
```

```
[1] " 1524157763907942"
```

Agentic behavior

| agent = reading tool + writing tool

Reading

```
1 chat$register_tool( tool(  
2   function(path) {  
3     dir(path)  
4   },  
5   name = "ls",  
6   description = "Lists the files in the given path",  
7   arguments = list(  
8     path = type_array(type_string())  
9   )  
10 ) )  
11  
12 chat$register_tool( tool(  
13   function(path) {  
14     readr::read_file(path)  
15   },  
16   name = "read_file",  
17   description = "Read the content of a file",  
18   arguments = list(  
19     path = type_array(type_string())  
20   )  
21 ) )
```

Writing

```
1 chat$register_tool( tool(  
2   function(content, path) {  
3     readr::write_file(content, path)  
4   },  
5   name = "write_file",  
6   description = "Write text in content to a file",  
7   arguments = list(  
8     content = type_array(type_string()),  
9     path = type_array(type_string())  
10   )  
11 ) )
```

Usage example

```
1 chat$chat( paste(
2   "Examine all of the files in the current directory – of the files ending in .qmd,",
3   "review their content. If they are about testthat then write a 1 paragraph summary",
4   "of their content to a file call testthat_slide_summary.md."
5 ) )
```

- [tool call] ls(path = ".")
 - #> data
 - #> imgs
 - #> Lec01.html
 - #> Lec01.pdf
 - #> Lec01.qmd
 - #> ...
- [tool call] read_file(path = "Lec01.qmd")
 - #> ---
 - #> title: "Welcome & Syllabus"
 - #> subtitle: "Lecture 01"
 - #> author: "Dr. Colin Rundel"
 - #> footer: "Sta 523 – Fall 2025"
 - #> ...
- [tool call] read_file(path = "Lec02.qmd")
 - #> ---
 - #> title: "Control Flow"
 - #> subtitle: "Lecture 02"
 - #> author: "Dr. Colin Rundel"
 - #> footer: "Sta 523 – Fall 2025"
 - #> ...
- [tool call] read_file(path = "Lec03.qmd")
 - #> ---
 - #> title: "Lists, Attributes, & S3"
 - #> subtitle: "Lecture 03"
 - #> author: "Dr. Colin Rundel"
 - #> footer: "Sta 523 – Fall 2025"
 - #> ...
- [tool call] read_file(path = "Lec04.qmd")
 - #> ---

Demo

OpenAI Codex & Claude Code