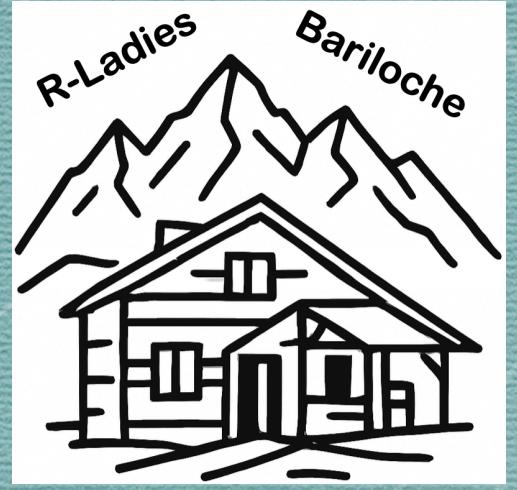


Para prepararte (MacBook):

1. Instala Ollama en tu ordenador (<https://ollama.com/>)
2. Ejecuta Ollama en tu ordenador ("ollama serve")
3. Descarga un modelo- en los ejemplos se usan "llama3.i" y "nomic-embed-text" ("ollama pull llama3:8b")
4. Ejecuta el modelo ("ollama run llama3:8b")
5. Chatea con el modelo

Asegúrate de tener:

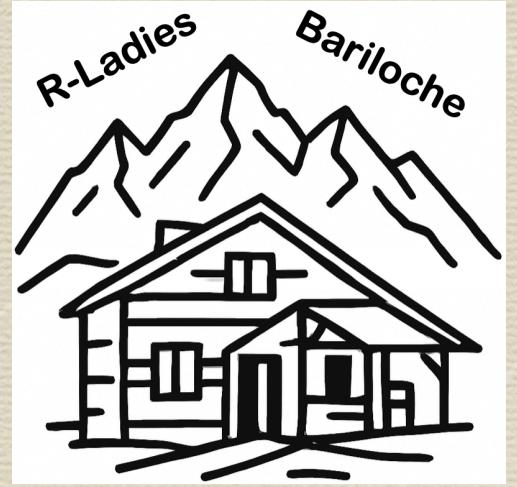
- * python3 instala (<https://www.python.org/downloads/>)
- * R instala (<https://cran.r-project.org/>)



Cómo ejecutar y personalizar un LLM para tu propio código

Integrating Applications and Code with LLMs

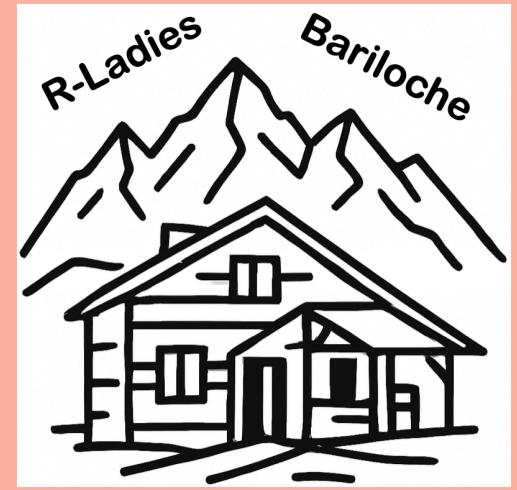
Kristin Tennessen
4 de Noviembre de 2025
R-Ladies Bariloche



Al final de esta charla usted...

At the end of this talk you will...

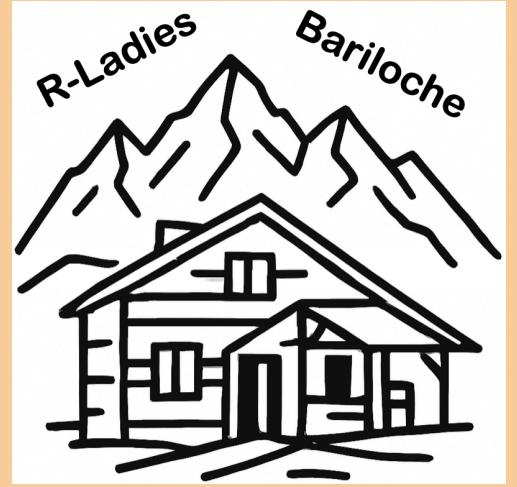
- * Ejecutará un LLM localmente
Run an LLM locally
- * Aprenderá diferentes maneras de integrar su LLM local en su código usando Python y R
Have learned different ways of integrating your local LLM into your code using Python and R
- * Comprenderá mejor las aplicaciones que usan Docker, React y Postgres, y cómo crearlas desde cero usando IA
Have a better understanding of applications using Docker, React, and Postgres, and how to build them from scratch using AI
- * Comprenderá mejor las necesidades de análisis de los miembros del grupo
Have a better understanding of the analysis needs of those of us in the group
- * Entenderá las limitaciones de la integración de los LLM
Understand the limitations of integrating LLMs



Quién es (who is) Kristin Tennessen?

- * Tengo 20 años de experiencia en investigación y tecnología en las biociencias
I have 20 years experience in research and technology in the biosciences
- * Amo la naturaleza y la vida outdoor
I love nature and the outdoors
- * Tengo un insaciable apetito por viajar
I have an insatiable appetite for travel
- * Estoy viviendo en Bariloche por 6 meses mientras mis hijos van a la escuela
I am living in Bariloche for 6 months while my kids go to school
- * Soy originalmente de la bahía de San Francisco en California
I am originally from the San Francisco Bay Area in California
- * Me está tomando tiempo aprender a hablar y comprender español
It is taking me awhile to learn how to speak and understand Spanish
- * Mi contacto:
WhatsApp: +1 925 979 8192
ktenness@gmail.com

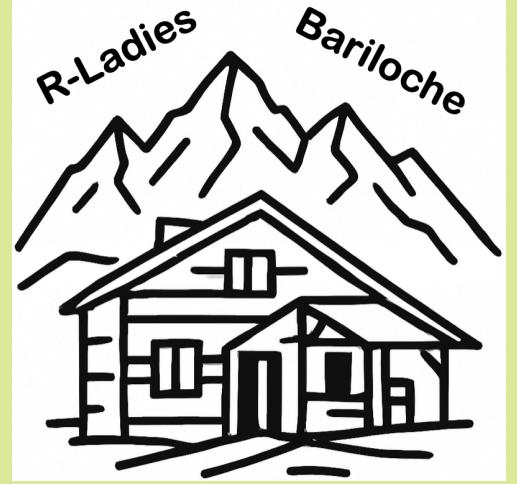




Horario de hoy

Schedule for today

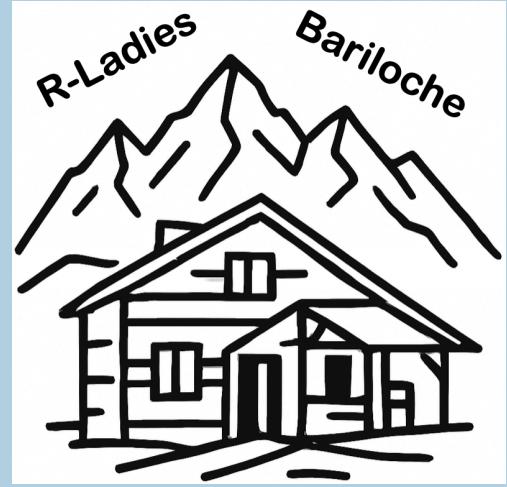
- * (20') Instalar Ollama y ejecutar un modelo
Install Ollama and run a model
- * (30') 3 ejemplos de integración de un LLM local en aplicaciones y código
3 examples of integrating a local LLM into applications and code
- * (20') Ejecutar un ejemplo en R localmente
Run an R example locally
- * (20') Crear una aplicación usando IA
Create an application using AI



Ética de la IA

AI Ethics

- * La IA plantea numerosos problemas éticos
There are many ethical issues with AI
- * Sin embargo, necesitamos una mayor diversidad de personas que desarrollen y trabajen con la IA
However, we need a larger diversity of humans developing and working with AI
- * Por eso nos centramos en la IA hoy en día
Hence, why we are focusing on AI today



Instala Ollama

Install Ollama

Ollama es una herramienta de código abierto que permite ejecutar modelos de lenguaje de gran tamaño (LLM) directamente en tu máquina local.

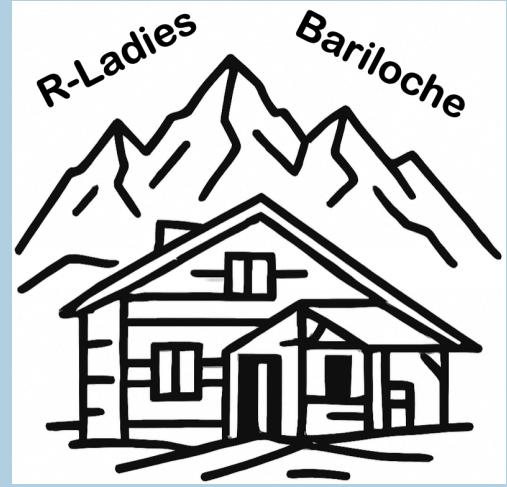
Ollama is an open-source tool that allows you to run large language models (LLMs) directly on your local machine.

<https://ollama.com/>

¿Por qué ejecutarlos localmente?

Why run locally?

- * Privacidad de datos
Data privacy
- * Personaliza los modelos según tus necesidades
Customize the models for you needs



Descarga y ejecuta un modelo

Download and run a model

1. Elige un modelo de esta lista / Choose a model from this list:

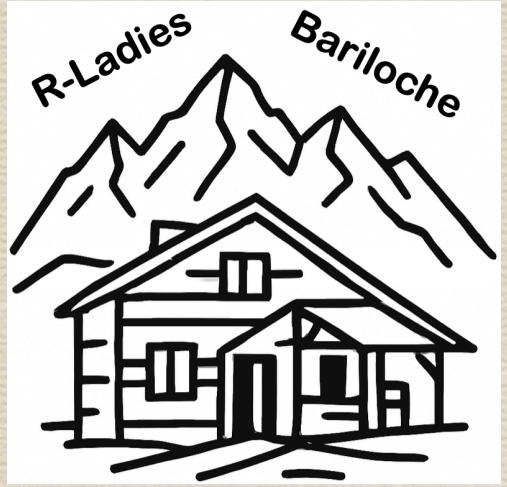
<https://ollama.com/search>

Les mostraré llama3.i (llama3:8b), un modelo de chat, y podrán integrarlo con un conjunto de datos en R.

I will demonstrate llama3.i (llama3:8b), a chat model, and you will be able to integrate it with a dataset into R.

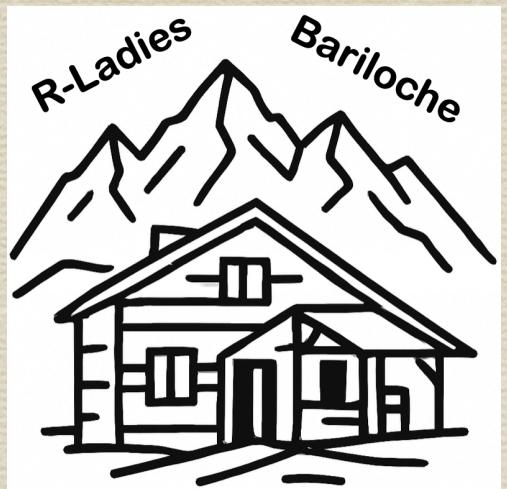
2. Obtén el modelo / Get the model: `ollama pull llama3.1:8b`

3. Ejecuta el modelo / Run the model: `ollama run llama3.1:8b`



Ejemplo 1: Freyoto (python)

- * Problema: Las fotos personales no se pueden buscar fácilmente en Google Photos
Problem: Personal photos aren't easily searchable on Google Photos
- * Solución: Crear una aplicación personalizada para consultar fotos. Utilizar un modelo de lenguaje natural (LLM) para convertir texto en consultas a la base de datos.
Solution: Build a personalized application for querying photos. Use an LLM to convert natural text into database queries.



Freyoto



Q 0,0



Q kristin in italy

Recent Searches

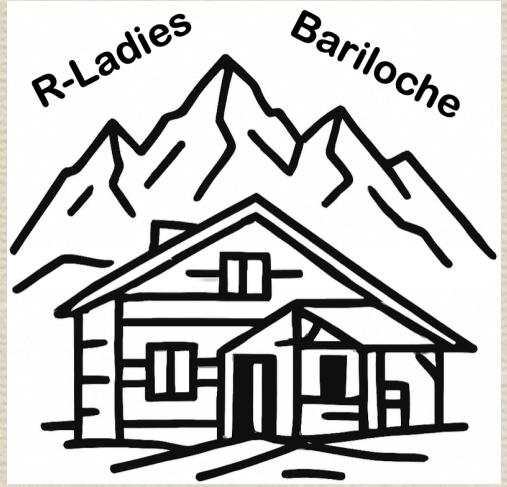
Photo Details



Date August 12, 2025

Location Unknown

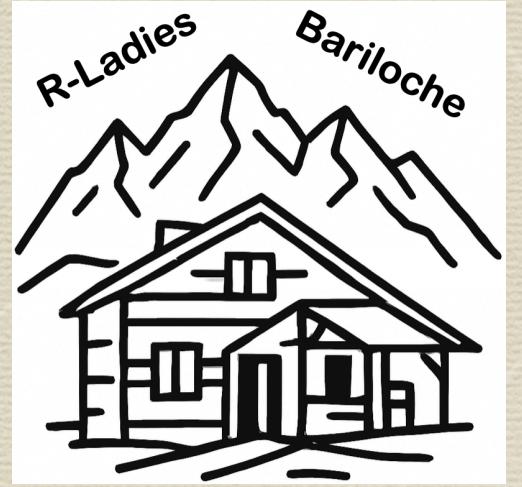
Description People



Ejemplo 1: Freyoto (python)

1. La búsqueda “kristin in italy” se envía al LLM
The search input “kristin in italy” is sent to the LLM
2. El LLM está configurado para devolver una consulta Django (SQL)
The LLM is configured to send back a django (SQL) query
3. Se consulta la base de datos y se devuelven las fotos coincidentes
The database is queried and the photo matches are returned

```
backend-1 | INFO: 192.168.65.1:54786 - "POST /graphql HTTP/1.1" 200 OK
backend-1 | initial photo queryset count: 252
backend-1 | kristin in italy
backend-1 | llm_response: Q(location__icontains="italy") & Q(face_names__icontains="kristin")
backend-1 | LLM filter: Q(location__icontains="italy") & Q(face_names__icontains="kristin")
backend-1 | INFO: 192.168.65.1:18102 - "POST /graphql HTTP/1.1" 200 OK
backend-1 | INFO: 172.19.0.4:47376 - "GET /2024%20July/IMG_6259.jpg HTTP/1.1" 200 OK
backend-1 | INFO: 172.19.0.4:47380 - "GET /2024%20July/IMG_6304.jpg HTTP/1.1" 200 OK
backend-1 | INFO: 172.19.0.4:47402 - "GET /2024%20July/IMG_6276.jpg HTTP/1.1" 200 OK
backend-1 | INFO: 172.19.0.4:47388 - "GET /2024%20July/IMG_6442.jpg HTTP/1.1" 200 OK
backend-1 | INFO: 172.19.0.4:47410 - "GET /2024%20July/IMG_6302.jpg HTTP/1.1" 200 OK
backend-1 | INFO: 172.19.0.4:47426 - "GET /2024%20July/IMG_6426.jpg HTTP/1.1" 200 OK
backend-1 | INFO: 172.19.0.4:47460 - "GET /2024%20July/IMG_5967.jpg HTTP/1.1" 200 OK
backend-1 | INFO: 172.19.0.4:47434 - "GET /2024%20July/IMG_6201.jpg HTTP/1.1" 200 OK
backend-1 | INFO: 172.19.0.4:47450 - "GET /2024%20July/IMG_6188.jpg HTTP/1.1" 200 OK
backend-1 | INFO: 172.19.0.4:47468 - "GET /2024%20July/IMG_5963.jpg HTTP/1.1" 200 OK
```



Freyoto: Arquitectura / Architecture

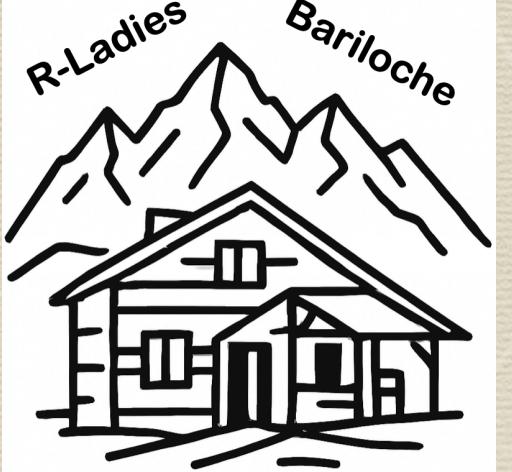
Frontend: React

Backend: Python/django

Base de datos / Database: Postgres

API: GQL (Graph Query Language)

Contenerización / Containerization: Docker



```
async def llama3_to_django_filter(search_query: str) -> str | None:
    prompt = f"""
You are an assistant that converts natural language photo search queries into Django ORM Q filter expressions for the Photo model.
The Photo model has fields: title, location, description, date_taken, and face_names (a comma-separated string of names, e.g. "kristin,danny" or "kristin").
- The family members are: parents: danny and kristin; kids (oldest to youngest): brock, sage, skye.
- "The oldest kid" means brock. "The youngest kid" means skye. "The oldest two kids" means brock and sage. "All kids" means brock, sage, and skye.
- To check if a photo contains a face, use Q(face_names__icontains="name").
- To check if a photo contains only one face (e.g. "only kristin"), use Q(face_names__icontains="kristin") & ~Q(face_names__contains=",") (no comma means only one name).
- The number of faces in a photo can be inferred from the number of commas in face_names: 0 commas = 1 face, 1 comma = 2 faces, 2 commas = 3 faces, etc.
- To check if a photo contains only two specific people (e.g. "only brock and skye"), match face_names exactly to "brock,skye" or "skye,brock" (since order may vary):
Q(face_names="brock,skye") | Q(face_names="skye,brock")
- If the search says a name without "only", find photos where that person appears, even if others are present: Q(face_names__icontains="name").
- If the search says a name with "only", find photos where that person(s) appears by so the the number of commas will be the number of persons minus 1.
- If the search contains portrait or people, search in description: Q(description__icontains="portrait")
- If the search contains alpine, search in description: Q(description__icontains="alpine")
- To check for a place (like "argentina"), use a fuzzy search: Q(location__icontains="argentina")
- For years, use Q(date_taken__year=YYYY)
- For activities or keywords, use Q(description__icontains="keyword") or Q(title__icontains="keyword")
```

Examples:

Query: all photos of kristin skiing from 2024

Output: Q(description__icontains="skiing") & Q(face_names__icontains="kristin") & Q(date_taken__year=2024)

Query: photos of only kristin by herself

Output: Q(face_names__icontains="kristin") & ~Q(face_names__contains=",")

Query: photos of kristin and danny and sage only

Output: Q(face_names="kristin,danny,sage") | Q(face_names="danny,kristin,sage") | Q(face_names="sage,kristin,danny") | Q(face_names="kristin,sage,danny") | Q(face_names="danny,sage,kristin") | Q(face_names="sage,danny,kristin")

Query: photos from argentina

Output: Q(location__icontains="argentina")

Query: photos of the oldest two kids

Output: Q(face_names__icontains="brock") & Q(face_names__icontains="sage") & ~Q(face_names__icontains="skye")

Query: photos of only oldest and youngest kid

Output: Q(face_names="brock") & Q(face_names="skye")

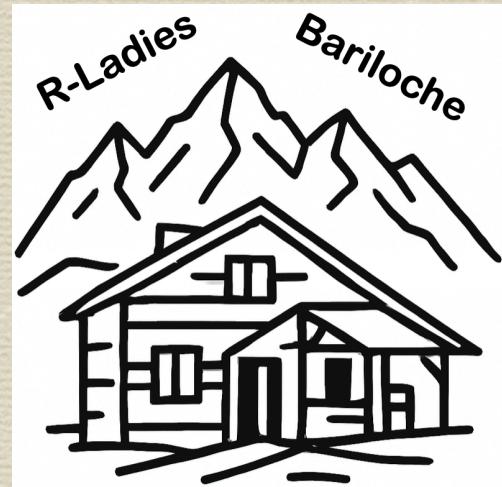
Now answer ONLY for the following query. Output ONLY the Django Q filter expression(s) for this query, and nothing else.

Query: {search_query}

Output:

```
"""\n    async with httpx.AsyncClient() as client:\n        response = await client.post(\n            "http://host.docker.internal:11434/api/generate",\n            json={\n                "model": "llama3:8b",\n                "prompt": prompt,\n                "stream": False\n            },\n            timeout=60,\n        )\n        result = response.json()\n        llm_response = result["response"].strip()\n        print("llm_response:", llm_response)\n        if llm_response.startswith("Q(") or llm_response.startswith("~Q("):\n            return llm_response\n        raise ValueError("No valid Q expressions found in LLM response")
```

<- El servidor Ollama se está ejecutando en el puerto 11434
<- Modelo utilizado llama3:8b
<- Configure el modelo con la prompt anterior

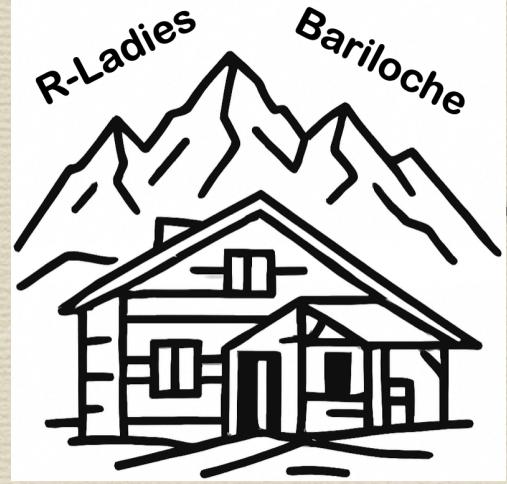


Ejemplo 2: ToFa (python)

- * Problema: Tengo más de 7000 publicaciones en mi blog personal. Al ser privado, no está indexado ni se puede buscar en Google. Quiero poder buscar en mi blog.
Problem: I have over 7,000 posts on my personal blog. Because it is private, it is not indexed and searchable via google. I want to search my blog.
- * Solución: Vectorizar el texto para que se pueda buscar mediante un modelo lógico de aprendizaje (LLM).
 1. Script de Python para descargar todas las publicaciones a archivos JSON locales.
 2. Ejecutar el modelo usando Llama.
 3. Usar una función de Python para consultar al LLM sobre los datos.

Solution: Vectorize the text so it can be searched via an LLM

 1. Python script to download all posts into local JSON files
 2. Run model using llama
 3. Use a python function to ask the LLM questions about the data



Archivo JSON para una entrada de blog

JSON file for a blog post

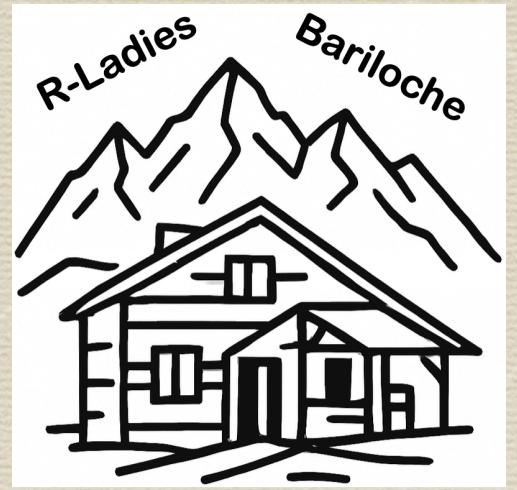
```
{  
  "id": "9131436707890795435",  
  "title": "Danny teaching Brock how to move the car",  
  "content": "<p class=\"mobile-photo\"><a href=\"https://blogger.googleusercontent.com/img/a/AVvXs\"><img src=\"https://blogger.googleusercontent.com/img/a/AVvXTtVNVOjP=s320\" border=\"0\" alt=\"\" id=\"BLOGGER_PHOTO_ID_7366326754\" /></a></p>So the driveway is free for kick ball",  
  "published": "2024-05-07T10:46:00-07:00",  
  "updated": "2024-05-07T10:46:15-07:00",  
  "url": "http://tortinafavola.blogspot.com/2024/05/how-to-move-car.html",  
  "status": "LIVE",  
  "author": {  
    "id": "08839414854593131318",  
    "displayName": "ktenness",  
    "url": "https://www.blogger.com/profile/08839414854591318"  
  },  
  "replies_total": "0",  
  "labels": [],  
  "custom_meta_data": "",  
  "location": {}  
}
```

vectorizar



vectorize

[0.36, 0.38, 0.4, ...]



Ej 2: Buscar en el blog

Ex 2: Search the blog

- * Utilice chromadb para insertar archivos JSON

Use chromadb to embed JSON files

- * ChromaDB es una base de datos vectorial. En lugar de guardar texto, números o filas como una base de datos SQL tradicional, ChromaDB almacena vectores — listas largas de números que representan el significado del texto.

ChromaDB is a vector database — a special kind of database built for AI applications. Instead of storing text, numbers, or rows like a normal SQL database, it stores vectors — long lists of numbers that represent the meaning of text.

- * Utilice las coincidencias vectoriales como contexto al consultar el LLM

Use the vector matches as a context when querying the LLM

- * El código está aquí / [Code is here](#)



Ej 2: Resultado de la consulta

Ex 2: Query results

```
ask_question("what did i write about mountains")
```

Blog post 1:

Refugio Otto Meiling on Cerro Tronador - The hike up the mountain was challenging and the wind almost blew down our tent on the summit....

Blog post 2:

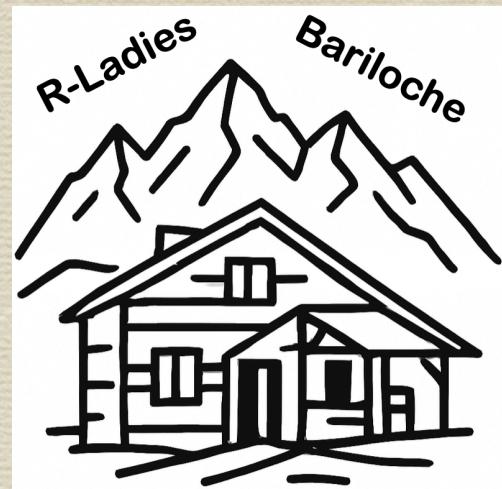
Danny carried Skye up to Mt Tallac - We arrived at the summit and had snacks. We could see all the way to...

Result of querying the LLM with the above context:

What a lovely collection of mountain-related posts you have! From these posts, it appears that you've written about mountains in various contexts:

1. ****Hiking and exploring**:** You've shared photos and stories about your hikes, including the scenic views, challenging climbs, and enjoyable moments with family and friends.
2. ****Mountain culture**:** Your posts have touched on aspects of mountain culture, such as appreciating cool breezes and finding joy in nature's beauty.

It seems that mountains hold a special place in your heart, and you've shared these experiences with your blog readers to inspire and share the joys of exploring the great outdoors!



Ejemplo 3: Conjunto de datos climáticos de ARG (R)

Example 3: ARG climate dataset (R)

Conjunto de datos climáticos / Climate dataset:

<https://ssl.smn.gob.ar/dpd/observaciones/estadisticas.txt>

Descripción / Description:

https://datos.gob.ar/dataset/smn-estadisticas-climaticas-normales/archivo/smn_8.1

Estadísticas climatológicas normales del período 1981-2010, valores medios mensuales de:

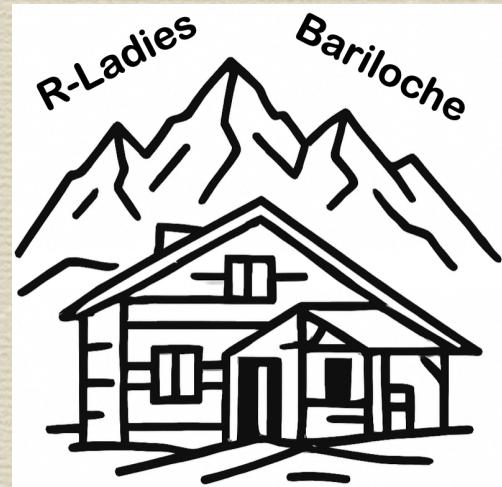
Temperatura, Temperatura máxima, Temperatura mínima, Humedad relativa,

Velocidad del viento, Nubosidad total,

Frecuencia de días con precipitación superior a 0.1 mm y Precipitación.

Normal climatological statistics for the period 1981-2010, average monthly values of: Temperature, Maximum temperature, Minimum temperature, Relative humidity, Wind speed, Total cloud cover, Frequency of days with precipitation greater than 0.1 mm and Precipitation.

Estación	Valor Medio de	Ene	Feb
LA QUIACA OBS.	Temperatura (°C)	12.8	12.5
LA QUIACA OBS.	Temperatura máxima (°C)	20.1	19.9
LA QUIACA OBS.	Temperatura mínima (°C)	7.4	7.0
LA QUIACA OBS.	Humedad relativa (%)	65.3	64.1
LA QUIACA OBS.	Velocidad del Viento (km/h)	6.9	7.1
LA QUIACA OBS.	Nubosidad total (octavos)	5.2	4.8
LA QUIACA OBS.	Precipitación (mm)	97.5	68.4
LA QUIACA OBS.	Frecuencia de días con Precipitación superior a 0.1 mm	15.9	12.2
ORÁN AERO	Temperatura (°C)	N/A	25.1
ORÁN AERO	Temperatura máxima (°C)	32.3	31.1



Ejemplo 3: Conjunto de datos climáticos de ARG (R)

Example 3: ARG climate dataset (R)

Ejecuta un script de R para resumir los datos ([el código está aquí](#)).

Run an R Script to summarize data ([code is here](#))

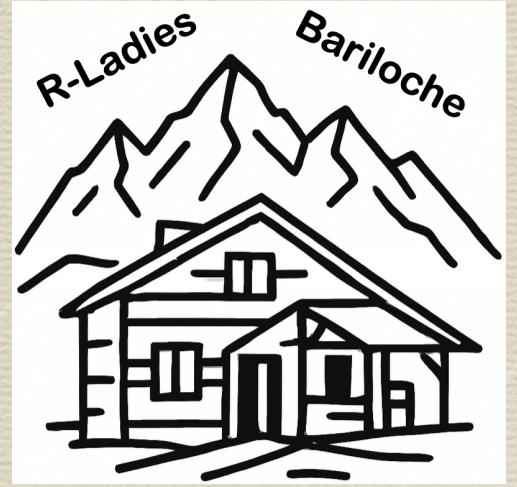
Resultados:

- * ¡Bariloche es frío!
- * ¡La Antártida (Base *) es más fría!

Results:

- * Bariloche is cold!
- * Antarctica (Base *) is colder!

Estación	TMAX_mean
<chr>	<dbl>
1 AEROPARQUE AERO	21.4
2 BAHÍA BLANCA AERO	22.2
3 BARILOCHE AERO	14.7
4 BASE BELGRANO II	-9.91
5 BASE ESPERANZA	-0.9
6 BASE MARAMBIO	-4.51
7 BASE ORCADAS	-0.183
8 BASE SAN MARTÍN	-1.35
9 BUENOS AIRES	22.7
10 CATAMARCA AERO	29.2



Ej 3: Conjunto de datos climáticos de ARG (R)

Ex 3: ARG climate dataset (R)

¿Podemos usar llama3 para generar un gráfico en R?

Can we use llama3 to generate an R plot?

Script de Python para generar código R

Python script to generate R code

Código R generado con llama3 que no funciona

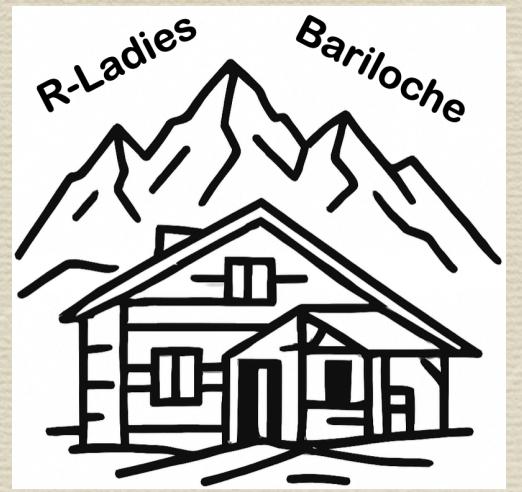
Generated R Code from llama3 that does not work

Código R corregido de llama3 para crear el gráfico

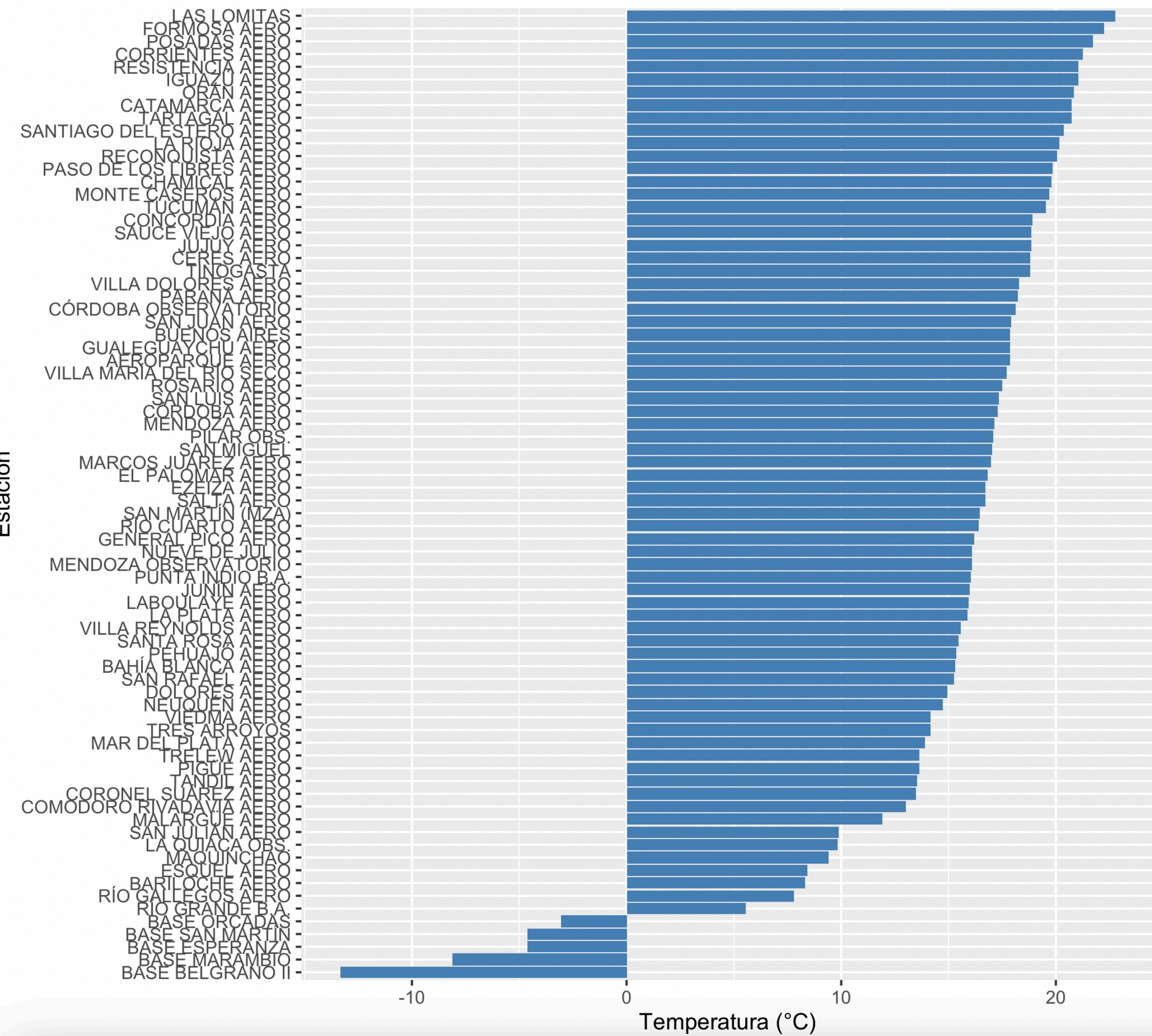
Fixed R Code from llama3 to create plot

Gráfico generado de temperaturas promedio

Generated plot of average temperatures



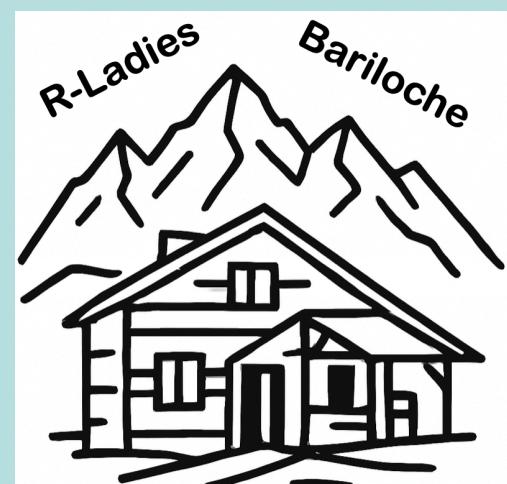
Temperatura promedio anual por estación (1981–2010)

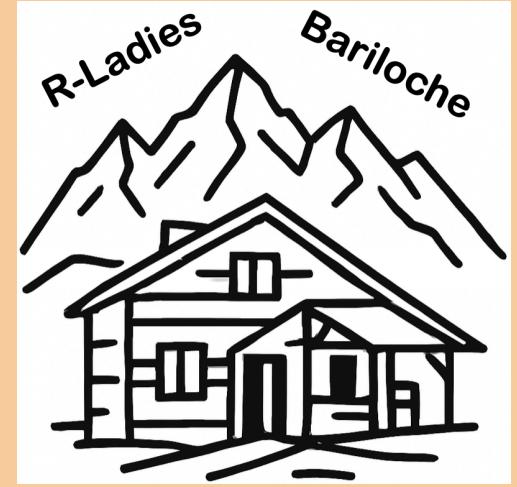


¡Práctica!: ¿Puedes generar un código R?

Hands-on: Can you generate R code?

1. Descarga el conjunto de datos climáticos de Argentina.
Download the Argentina climate dataset
2. Copia el script de Python para generar el código R y modifica la instrucción LLM para adaptarla a tu pregunta de análisis.
Copy the python script for generating R code and change the LLM prompt for your analysis question
3. Al ejecutar el script de Python, ¿se ejecuta el código R?
Run the python script, does the R code run?





Creación de una solicitud con LLMs

Creating an application with LLMs

1. Crea el diseño con [Stitch](#)

Create the design with [Stitch](#)

2. Descarga el archivo HTML generado e impórtalo a [claude.ai](#)

Download the generated html file and input it into [claude.ai](#)

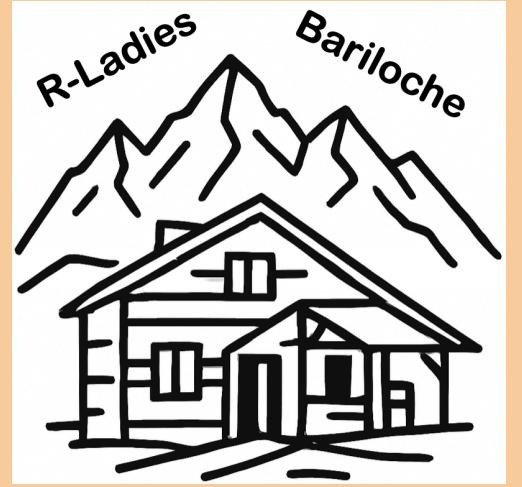
3. Copia los archivos a tu ordenador y ejecuta el programa.

Copy the files locally and run

4. Utiliza un chat (no uses el chat local, ies muy lento!) para solucionar los errores.

Use a chat (not local, it is too slow!) to fix the errors

Solicitar acceso al repositorio en GitHub / Request access to the repo in GitHub:
<https://github.com/ktleness/taptales>



El agua de Bariloche tiene un sabor delicioso!
Problema: Pero no en todas partes del mundo.

Solución:
Una aplicación de
crowdsourcing
para rastrear en
qué partes del
mundo el agua
tiene buen,
regular o
mal sabor.

TapTales

Account

Rate Your Tap Water
Help the community by sharing your experience.

Hi there! I'm the TapTales bot. How would you rate your current tap water? Good, Okay, or Gross?

It's pretty good today!

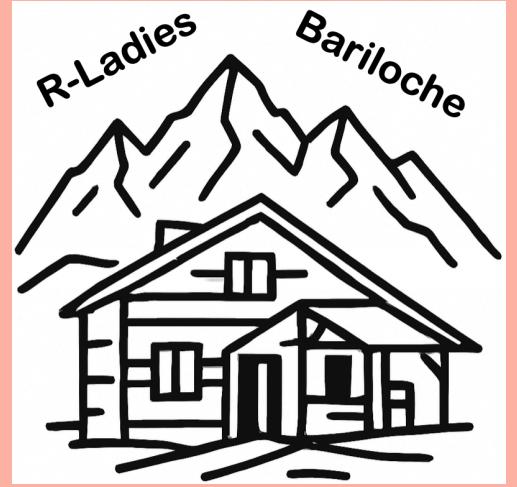
Great to hear! Your "Good" rating for Brooklyn, NY has been recorded. Thanks for contributing!

Mapa de aguas clasificadas en todo el mundo

Brooklyn, NY
Overall: Okay

75% 15% 10%

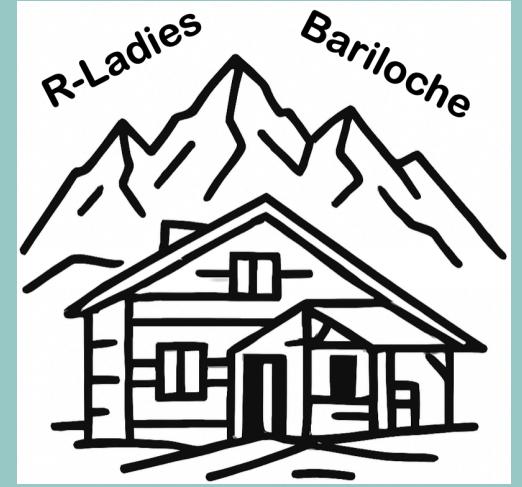
Type your rating... ➤



Conclusiones

Takeaways

- * La IA, ahora mismo, no puede reemplazarte
AI, right now, can not replace you
- * La IA, ahora mismo, puede hacer tareas no especializadas. Entonces tu puedes concentrarte en recolectar la información correcta y en realizar las preguntas correctas
AI, right now, can do menial tasks so you can concentrate on collecting the right data and asking the right questions
- * La IA está mejorando, rápido. Hagámosla mas ética comprendiéndola y usándola.
AI is getting better, fast. Let's contribute to making it more ethical by understanding it and using it.



Discusión

Discussion

¿Pueden los LLM locales ayudarte a resolver problemas técnicos?

Can local LLMs assist you in solving technical problems?

Seguimiento de la reunión anterior

Follow up from last meeting

Seguimiento de la reunión anterior: Ideas para una cultura laboral más inclusiva, recursos de mi trabajo en una organización sin ánimo de lucro que promueve el ciclismo y lucha contra el racismo. Estas listas ayudan a identificar áreas de mejora y cómo mejorarlas.

Follow up from last meeting: Ideas for more inclusive working culture, resources from my work on a non-profit bicycling advocacy group addressing racism. These lists help identify areas of improvement, and how to improve it:

- Características frente a antídotos / Characteristics vs antidotes: <https://www.berkeleypublicschools.org/wp-content/uploads/2020/07/White-Supremacy-Culture-Summary.pdf>
- <https://www.whitesupremacyculture.info/uploads/4/3/5/7/43579015/okun - white sup culture 2020.pdf>
- <https://www.cacgrants.org/assets/ce/Documents/2019/WhiteDominantCulture.pdf>