Automated ADRG Generation with LLM Utilities

Pilot 5 DatasetJSON Approach for FDA Submissions

https://github.com/RConsortium/submissions-pilot5-datasetjson/tree/main?tab=readme-ov-file

What is an ADRG?

- A detailed guide that explains:
    - Dataset structures (ADaM, SDTM)
    - Variable derivations and calculations
    - Analysis methods and assumptions
    - Expected output tables and results
- Required for FDA submissions to ensure reviewers understand the analysis logic

Traditional ADRG Workflow (Without LLMs)

Content / Diagram:

1. Extract dataset metadata from ADaM/SDTM
2. Generate descriptive tables manually (e.g., variable lists, datasets, derivations)
3. Write narrative text explaining:
    - Dataset derivations
    - Analysis assumptions and methods
    - Relationships between variables
4. Compile everything in **R Markdown or Quarto** → Output PDF

Pilot 5 LLM Automation Overview

Content / Diagram:

Goal: Automate repetitive and structured parts of ADRG using LLMs.

Pipeline Overview:

1. **Prepare structured input data**: JSON, dataset metadata
2. **Create prompts for LLM**: instructions to generate narrative text
3. **Call LLM** → generate draft text for ADRG sections
4. **Extract structured tables** (if LLM outputs CSV or JSON)
5. **Insert content into Quarto/R Markdown** → compile ADRG

Core LLM Utility Functions

Function	Purpose	Notes
`infer_provider_from_model(model)`	Automatically determines which LLM provider to use based on model name	e.g., GPT → OpenAI, Claude → Anthropic
`create_chat_object(provider, model, api_key)`	Initializes a chat session with the LLM	Handles API configuration
`retry_with_exponential_backoff(call)`	Re-tries calls when network or API errors occur	Avoids failures due to transient errors
`llm_call(prompt, model)`	Main function to generate narrative text from LLM	Returns plain text ready for ADRG
`llm_extract_structured(prompt, model)`	Extracts structured outputs (CSV/JSON) from LLM	Useful for tables and variable lists

Logging Utilities

Function	Purpose	Notes
`ensure_logs_dir()`	Ensures a `logs/` folder exists for storing LLM call logs	Creates folder if missing
`log_llm_call(prompt, model, response)`	Saves time stamped logs of each LLM call	Provides audit trail for regulatory purposes

Parsing Tables from LLM Output

Function	Purpose	Notes
`convert_to_dataframe(csv_string)`	Converts LLM-generated CSV text into an R data frame	Strips markdown code fences, headers, and extra text

Logical Flow of Automated ADRG Generation

Step-by-Step Workflow:

1. **Input Preparation:**
    - ADaM/SDTM metadata
    - JSON dataset descriptions
    - Prompt templates for LLM
2. **LLM Narrative Generation:**
    - Call `llm_call(prompt, model)`
    - Log call using `log_llm_call(prompt, model, response)`
3. **Structured Table Extraction (Optional):**
    - Use `llm_extract_structured()` if the LLM returns CSV/JSON
    - Convert table with `convert_to_dataframe()`
4. **Document Assembly:**
    - Insert narrative and tables into Quarto/R Markdown
    - Compile ADRG into PDF/HTML
5. **Review and QA:**
    - Human review for accuracy and compliance

Benefits of LLM Automation

- Reduces **manual effort** for narrative sections
- Ensures **consistent phrasing** across datasets and derivations
- Allows **programmatic extraction of tables** for reproducibility
- Generates an **audit trail** through logging
- LLM assists drafting, **but human review remains critical**

Summary

Aspect	Traditional ADRG	LLM-Assisted ADRG
Effort	Manual, repetitive, time-consuming	Automated drafting of narratives and tables
Consistency	Prone to variation	Consistent phrasing and formatting
Reproducibility	Harder to track	Logs and structured outputs ensure reproducibility
QA	Required at every step	Still required, but LLM reduces workload

Conclusion:

- Pilot 5 ADRG automation demonstrates **how LLMs can assist regulatory document creation**.
- Narrative and table generation can be automated, but human oversight ensures **regulatory compliance and quality**.