# Assignment 5 – Scraping Government Data from GovInfo.gov

This project demonstrates automated collection of government documents using the GovInfo.gov public API. The task was to extract JSON and CSV listings of U.S. Congressional publications, filter valid PDF links, and download the documents using R.

## Workflow Summary

| Step | Description | R Packages Used |
|---|---|---|
| 1 | Environment setup and package installation | base R, utils |
| 2 | Accessing and reading JSON/CSV data from GitHub (GovInfo exports) | rjson, jsonlite, readr |
| 3 | Data cleaning and selection of relevant columns | dplyr |
| 4 | Creating custom download function with error handling and rate limiting | purrr, magrittr |
| 5 | Testing single-file and batch downloads | purrr, Sys.time() |
| 6 | Saving PDFs to local directory | base R file functions |

**Key Results**
- Total valid downloadable files: 693
- Test downloads:
  • Single file: ■ Success in 14.6 seconds
  • Batch (5 files): ■ Success in 18.1 seconds
  • Bulk (10 files): ■ Success in 33.5 seconds
- Average download size: ~230 KB per file
- All files stored in local folder: **govinfo_pdfs**.

**Challenges**
- Some files in the dataset lacked pdfLink entries.
- Occasional slow responses depending on server load.
- Maintaining random pauses was essential to avoid rate limits.

**Lessons Learned**
- JSON APIs are powerful for structured access to government data.
- Automating downloads with purrr::map_chr() is efficient and concise.
- Responsible scraping practices (sleep intervals, error handling) prevent blocking.
- This workflow can be extended for longitudinal document analysis or text mining.

**Conclusion**
The GovInfo.gov data scraping exercise successfully demonstrates the application of R for large-scale, reproducible, and ethical government data collection. The script developed can be scaled to continuously monitor Congressional reports, treaties, or committee hearings in near real-time.