Statistics Canada

Modernisation of Retail Services Price Index

Delivering insight through data for a better Canada

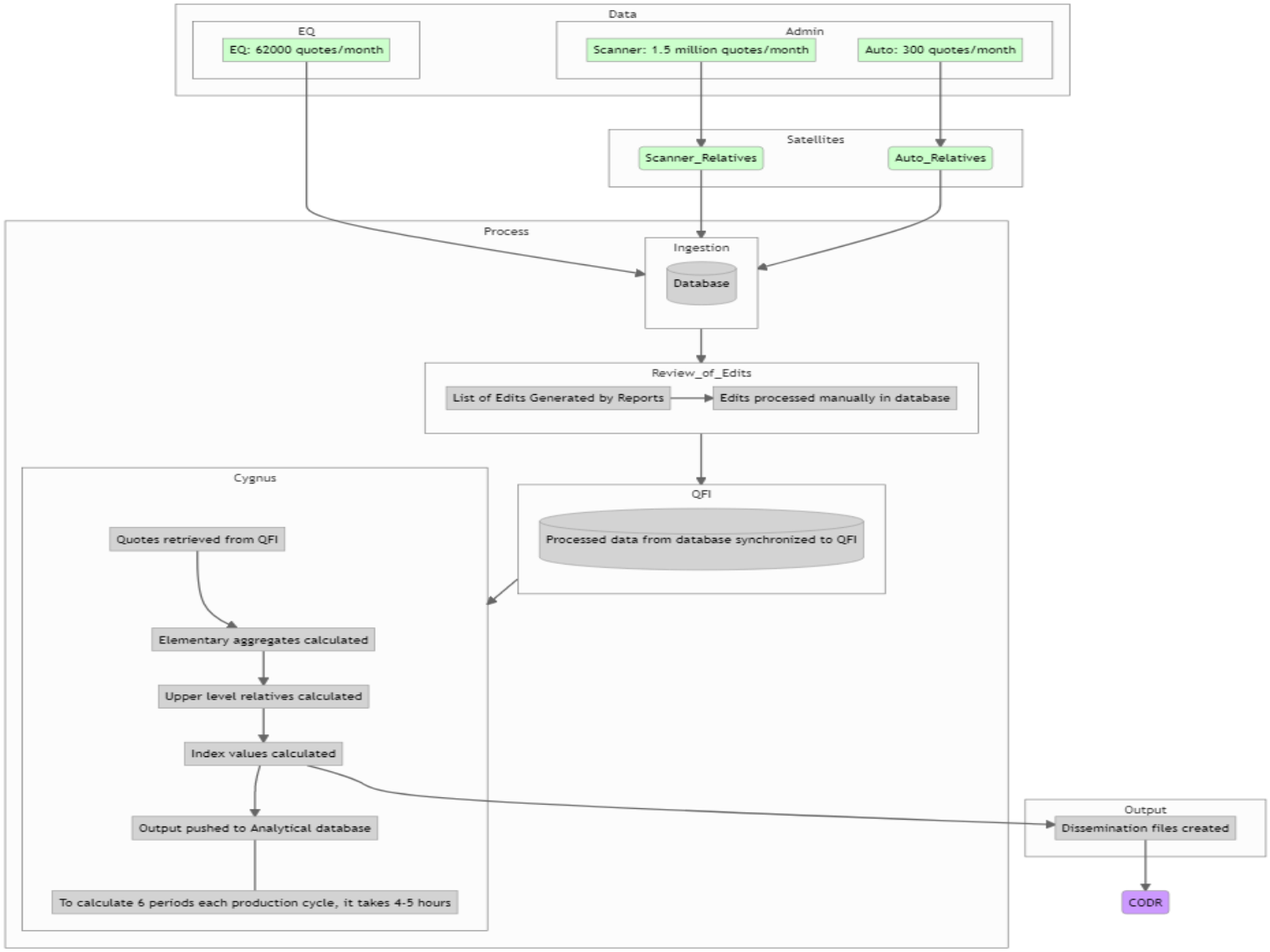
Retail Services Price Index: Modernization

- Starting two years ago, the Retail Services Price Index commenced a modernization project to make it more relevant and efficient
- Modernization was tackled on two fronts:
  - Data
  - Production Systems

Retail Services Price Index: Data Modernization

- Effort was put into improving the quality of data
- Two major outcomes:
  - Incorporation of Scanner data to improve the quality of our index in NAICS 44511: Supermarkets and other grocery (except convenience) stores
    - Around 1.5 million quotes per month
  - Incorporation of an admin source to measure NAICS 44111: New car dealers in our next basket which was previously not measured in the Retail Services Prices Index
    - About 300 quotes per month
- In addition, products have been NAPCS coded to allow for more detailed analysis at product level indices when needed

Retail Services Price Index: System Modernisation - Flow Chart of Current Process



Retail Services Price Index: Current Process vs New Process

Current Process

Maintenance
- Complex ecosystem with multiple systems
- Changes and improvements require help of IT

Not Flexible
- Process is rigid
- Need to follow pre-defined steps
- Very cumbersome to adapt and implement new baskets

Inefficient
- Processing of data is manual
- Calculation of index takes a long time i.e. 4 to 5 hours for each run

Cost
- Costly to maintain due to multiple systems

New Process

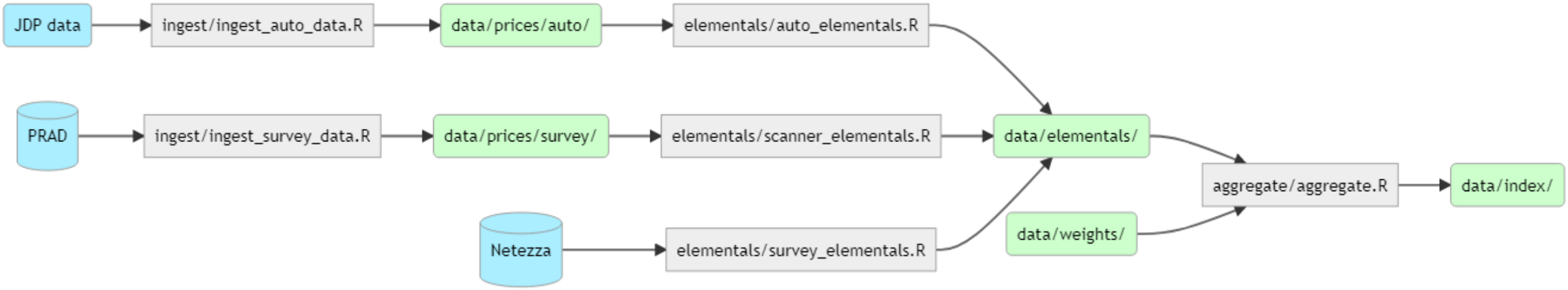
Maintenance
- Simpler ecosystem
- Changes can be done without IT support

Flexible
- Easy to adapt
- New processes can be added easily

Efficient
- Much faster to calculate: minutes instead of hours
- Processing of data can be automated

Cost
- Uses open-source software
- Less IT resources needed

Retail Services Price Index: New Process Flow Chart



Retail Services Price Index: Demo

automated price indexes › rspi

rspi
Project ID: 5743

Star 0 | Fork 2

96 Commits | 15 Branches | 0 Tags | 2 MB Project Storage

An automated process to make the Retail Services Price Index

master | rspi / | + | Find file | Web IDE | Clone

A Merge branch 'contributing-guidelines' into 'master'
Ambuj Dewan authored 1 day ago | bf75d401

README | CHANGELOG | CONTRIBUTING | CI/CD configuration | Add LICENSE | Configure Integrations

Name	Last commit	Last update
aggregate	Added mention of contributions to relevant ...	1 week ago
analysis	Corrected export location.	1 day ago
data	Added contributions subfolder to the data f...	3 weeks ago
disseminate	Forced readline and removed include.only fo...	1 week ago
elementals	Added mention of contributions to relevant ...	1 week ago
ingest	Survey query now uses FinalValue to reflect ...	6 days ago
sample_management	substitute mardown for Rmarkdown	6 months ago
tests	Added data dissemination directory	2 months ago
.gitignore	Initial commit	8 months ago
.gitlab-ci.yml	Updated E2E test to use latest version of pric...	2 months ago
CHANGELOG	Add CHANGELOG	8 months ago
CONTRIBUTING.md	Revised contributing guidelines	1 day ago
DATA.md	Updated readme.	5 months ago
README.md	Updated dependencies and package installa...	2 months ago

master | rspi / | +

Switch branch/tag

Search branches and tags

Branches

- quarterly-contributions
- ✓ master
- parallel-run
- parallel-run-chain
- parallel-run-documentation
- dev

automated price indexes › rspi › Issues

Open 14 | Closed 45 | All 59 | Edit issues | New issue

Search or filter results... | Created date

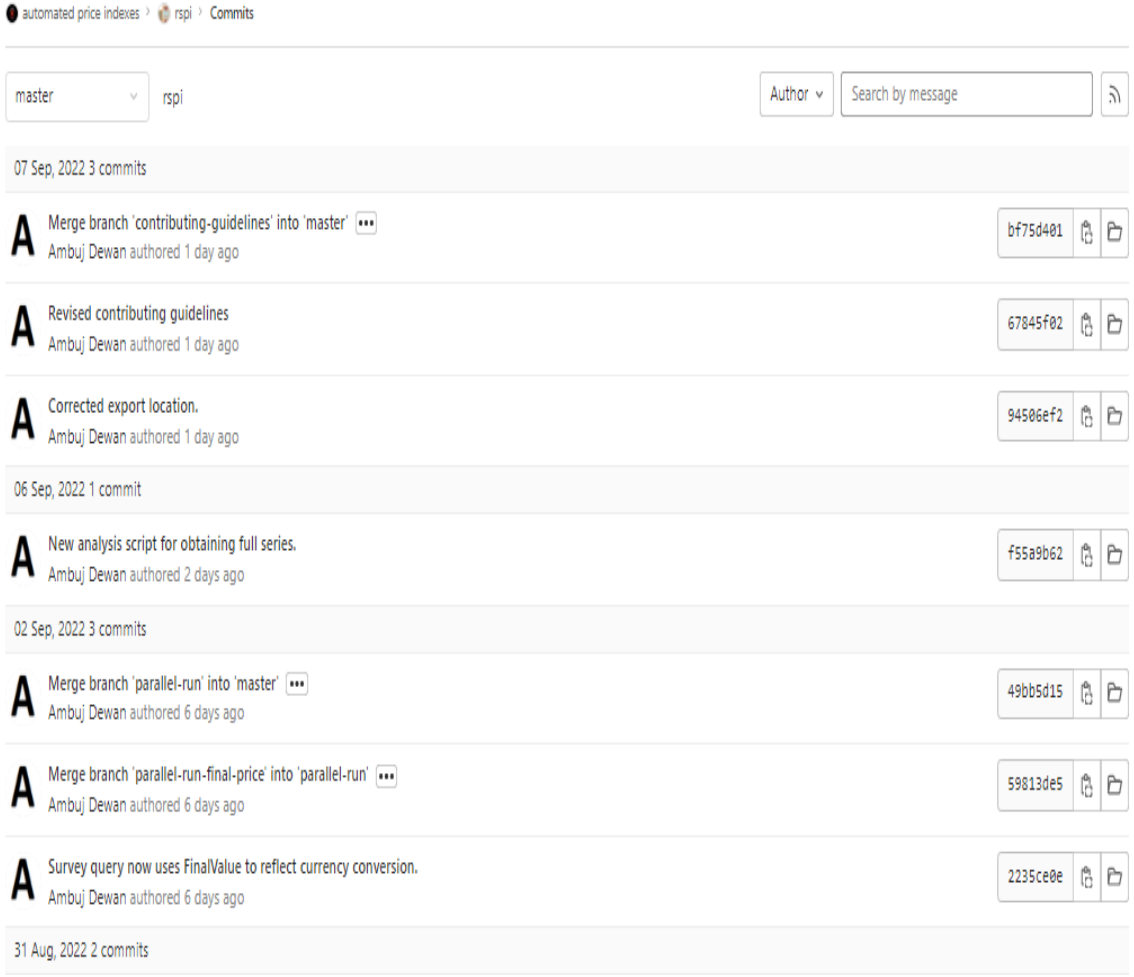
(Potentially) replace substring method for stripping trailing slashes with a regex-based method (i.e. gsub)
#54 · created 2 weeks ago by Ambuj Dewan Dec 1, 2022 enhancement
updated 2 weeks ago

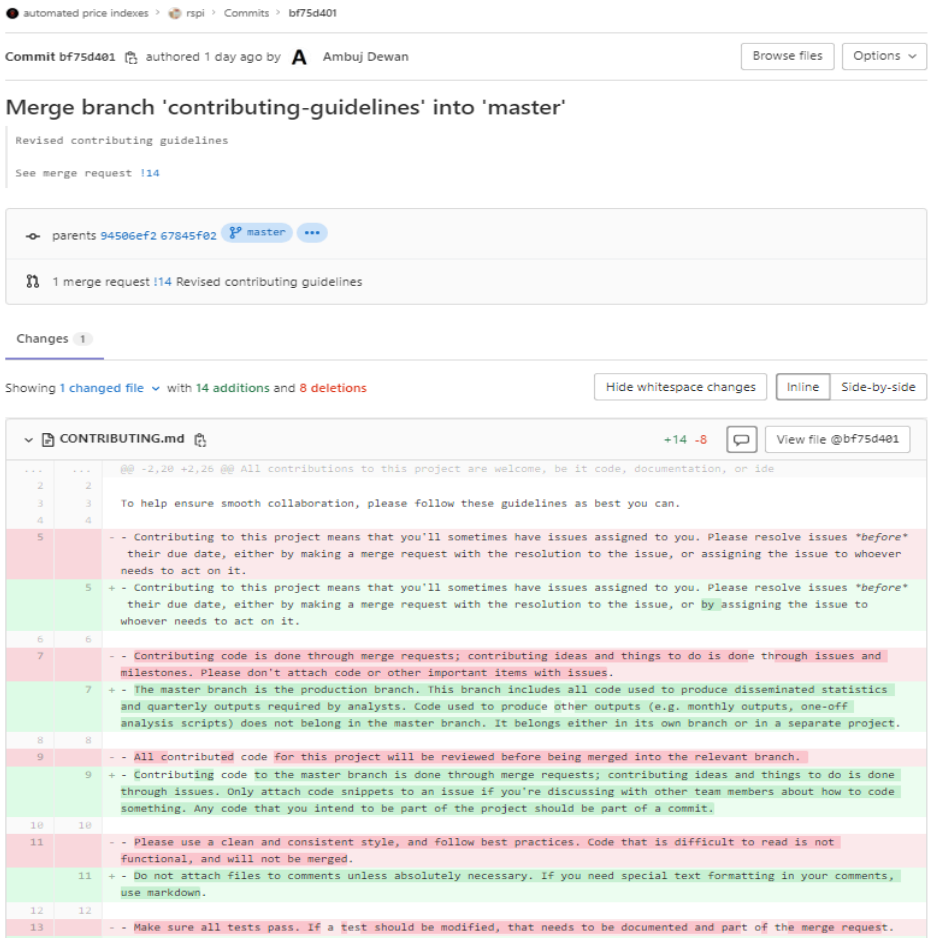
Allow both empty strings and unsupplied arguments from the command line
#53 · created 2 weeks ago by Ambuj Dewan Sep 30, 2022 enhancement
updated 2 weeks ago

Base period flexibility
#52 · created 3 weeks ago by Ambuj Dewan Dec 23, 2022 enhancement
updated 3 weeks ago

Make auto data toggle Boolean
#49 · created 3 weeks ago by Ambuj Dewan Sep 30, 2022 enhancement
updated 3 weeks ago

Retail Services Price Index: Demo





Retail Services Price Index: Demo

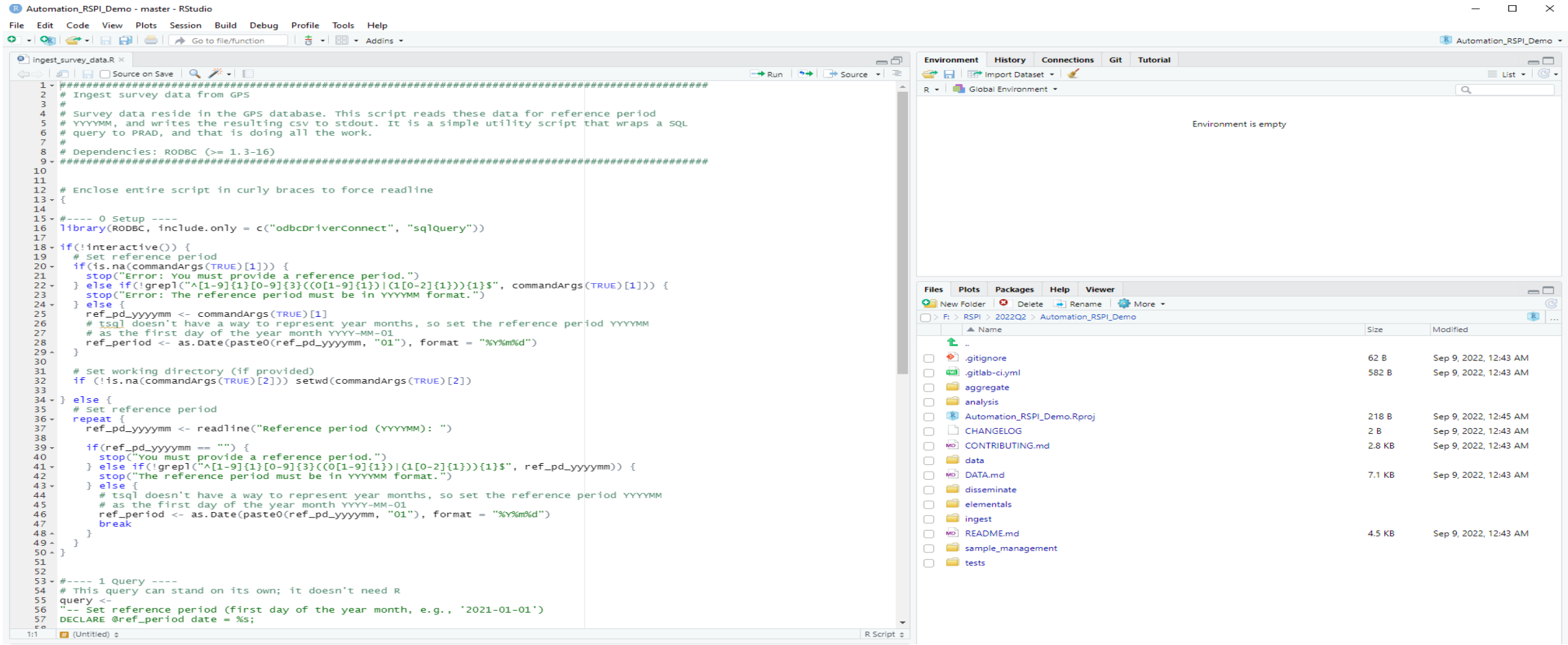
› RSPI › 2022Q2

ne	Date modified	Type	Size

Name	Date modified	Type
Automation_RSPI_Demo	9/8/2022 8:45 PM	File folder

Name	Date modified	Type	Size
aggregate	9/8/2022 8:43 PM	File folder	
analysis	9/8/2022 8:43 PM	File folder	
data	9/8/2022 8:44 PM	File folder	
disseminate	9/8/2022 8:44 PM	File folder	
elementals	9/8/2022 8:44 PM	File folder	
ingest	9/8/2022 8:44 PM	File folder	
sample_management	9/8/2022 8:44 PM	File folder	
tests	9/8/2022 8:45 PM	File folder	
.gitignore	9/8/2022 8:43 PM	Text Document	1 KB
.gitlab-ci.yml	9/8/2022 8:43 PM	YML File	1 KB
Automation_RSPI_Demo	9/8/2022 8:45 PM	R Project	1 KB
CHANGELOG	9/8/2022 8:43 PM	File	1 KB
CONTRIBUTING.md	9/8/2022 8:43 PM	MD File	3 KB
DATA.md	9/8/2022 8:43 PM	MD File	8 KB
README.md	9/8/2022 8:43 PM	MD File	5 KB

Retail Services Price Index: Demo




```
##############################################################################################
# Ingest survey data from GPS
#
# Survey data reside in the GPS database. This script reads these data for reference period
# YYYYMM, and writes the resulting csv to stdout. It is a simple utility script that wraps a SQL
# query to PRAD, and that is doing all the work.
#
# Dependencies: RODBC (>= 1.3-16)
##############################################################################################


# Enclose entire script in curly braces to force readline
{

#---- 0 Setup ----
library(RODBC, include.only = c("odbcDriverConnect", "sqlQuery"))

if(!interactive()) {
  # Set reference period
  if(is.na(commandArgs(TRUE)[1])) {
    stop("Error: You must provide a reference period.")
  } else if(!grepl("^[1-9]{1}[0-9]{3}((0[1-9]{1})|(1[0-2]{1})){1}$", commandArgs(TRUE)[1])) {
    stop("Error: The reference period must be in YYYYMM format.")
  } else {
    ref_pd_yyyymm <- commandArgs(TRUE)[1]
    # tsql doesn't have a way to represent year months, so set the reference period YYYYMM
    # as the first day of the year month YYYY-MM-01
    ref_period <- as.Date(paste0(ref_pd_yyyymm, "01"), format = "%Y%m%d")
  }

  # Set working directory (if provided)
  if (!is.na(commandArgs(TRUE)[2])) setwd(commandArgs(TRUE)[2])

} else {
  # Set reference period
  repeat {
    ref_pd_yyyymm <- readline("Reference period (YYYYMM): ")

    if(ref_pd_yyyymm == "") {
      stop("You must provide a reference period.")
    } else if(!grepl("^[1-9]{1}[0-9]{3}((0[1-9]{1})|(1[0-2]{1})){1}$", ref_pd_yyyymm)) {
      stop("The reference period must be in YYYYMM format.")
    } else {
      # tsql doesn't have a way to represent year months, so set the reference period YYYYMM
      # as the first day of the year month YYYY-MM-01
      ref_period <- as.Date(paste0(ref_pd_yyyymm, "01"), format = "%Y%m%d")
      break
    }
  }
}


#---- 1 Query ----
# This query can stand on its own; it doesn't need R
query <-
"-- Set reference period (first day of the year month, e.g., '2021-01-01')
DECLARE @ref_period date = %s;
```