Module 1: Tabular Data

Working with larger-than-RAM data using duckdbfs

ESPM 288

Introduction

In this module, we will explore high-performance workflows for tabular data. We will use `duckdbfs` to work with datasets that are larger than available RAM by leveraging DuckDB's streaming and remote file access capabilities.

Case Study: Global Supply Chains

We will be working with EXIOBASE 3.8.1, a global Multi-Regional Input-Output (MRIO) database. This dataset tracks economic transactions between sectors and regions, along with their environmental impacts (emissions, resource use, etc.).

Data description: - **Coverage**: 44 countries + 5 rest-of-world regions. - **Timeframe**: 1995–2022. - **Content**: Economic transactions (Z matrix), final demand (Y matrix), and environmental stressors (F matrix). - **Format**: Cloud-optimized Parquet, partitioned by year and matrix type.

Setup

```
library(duckdbfs)
library(dplyr)
```

```
Attaching package: 'dplyr'
```

```
The following objects are masked from 'package:stats':

    filter, lag
```

```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
```

Exercise 1: connecting to remote data

We can open the entire dataset without downloading it using `open_dataset()`. The data is hosted on Source Cooperative. The `**` pattern allows recursive scanning of the partitioned parquet files.

```
# Remote S3 path to EXIOBASE 3 (Source Cooperative)

duckdbfs::duckdb_secrets(
    key = "",
    secret = "",
    endpoint = "s3.amazonaws.com",
    region = "us-west-2"
)
```

```
[1] 1
```

```
s3_url <- "s3://us-west-2.opendata.source.coop/youssef-harby/exiobase-3/4588235/parquet/year=

# Open the dataset lazily
exio <- open_dataset(s3_url)

# View the schema (column names and types) without reading data
glimpse(exio)
```

```
Rows: ??
Columns: 8
Database: DuckDB v1.3.2 [alexj@Darwin 24.6.0:R 4.3.3/:memory:]
$ stressor <chr> "Value Added", "Value Added", "Value Added", "Value Added", "~
$ region   <chr> "AT", "AT", "AT", "AT", "AT", "AT", "AT", "AT", "AT", "AT", "~
$ sector   <chr> "Cultivation of wheat", "Cultivation of cereal grains nec", "~
$ value    <dbl> 183.1118891, 402.2305799, 830.2127384, 101.9705426, 31.763189~
$ unit     <chr> "M.EUR", "M.EUR", "M.EUR", "M.EUR", "M.EUR", "M.EUR", "M.EUR"~
```

```
$ year     <dbl> 1995, 1995, 1995, 1995, 1995, 1995, 1995, 1995, 1995, 1995, 1~
$ format   <chr> "ixi", "ixi", "ixi", "ixi", "ixi", "ixi", "ixi", "ixi", "ixi"~
$ matrix   <chr> "F_impacts", "F_impacts", "F_impacts", "F_impacts", "F_impact~
```

Exercise 2: Efficient Filtering

The dataset is large. We should filter *before* collecting any data into R. US 2022 filter

```
exio |>
    filter(year == 2022, region == "US") |>
    head() |> # view the first 6 rows
    collect()
```

```
# A tibble: 6 x 8
  stressor    region sector                      value unit   year format matrix
  <chr>       <chr>  <chr>                       <dbl> <chr> <dbl> <chr>  <chr>
1 Value Added US     Cultivation of paddy rice    750. M.EUR  2022 ixi    F_imp~
2 Value Added US     Cultivation of wheat        2019. M.EUR  2022 ixi    F_imp~
3 Value Added US     Cultivation of cereal gra~  7355. M.EUR  2022 ixi    F_imp~
4 Value Added US     Cultivation of vegetables~ 26878. M.EUR  2022 ixi    F_imp~
5 Value Added US     Cultivation of oil seeds    5003. M.EUR  2022 ixi    F_imp~
6 Value Added US     Cultivation of sugar cane~   290. M.EUR  2022 ixi    F_imp~
```

What happened globally in 1995? What countries and sectors were the biggest emitters?

```
exio |>
    filter(year == 1995) |>
    group_by(region) |>
    summarize(total_emissions = sum(.data$value, na.rm = TRUE), .groups = "drop") |>
    arrange(desc(total_emissions)) |>
    head(20) |>
    collect()
```

```
# A tibble: 20 x 2
   region total_emissions
   <chr>            <dbl>
 1 WA             3.10e16
 2 CN             2.53e16
```

```
 3 IN             8.19e15
 4 WF             8.17e15
 5 US             7.28e15
 6 WL             4.97e15
 7 BR             4.81e15
 8 WE             1.97e15
 9 MX             1.92e15
10 WM             1.82e15
11 RU             1.76e15
12 IE             1.35e15
13 FR             1.16e15
14 ID             1.08e15
15 DE             1.01e15
16 CA             7.94e14
17 GB             7.78e14
18 PL             7.15e14
19 AU             7.15e14
20 ES             7.02e14
```

Task Construct a query to find the top 5 sectors in the US by CO2 emissions in 2022. Remember to check the column names in `exio` to find the appropriate emissions flow.

```
#a query written based on the task above, used the chatbot and the info provide in the glimps

exio |>
  distinct(stressor) |>
  collect()
```

```
# A tibble: 124 x 1
   stressor
   <chr>
 1 Domestic Extraction Used - Crop and Crop Residue
 2 Domestic Extraction Used - Iron Ore
 3 Acidification endpoint | ILCD recommended CF | Change in potentially not occ~
 4 Eutrophication terrestrial midpoint | ILCD recommended CF | Accumulated Exce~
 5 Damage to Ecosystem Quality caused by the combined effect of acidification a~
 6 Terrestrial ecotoxicity (TETP100) | Problem oriented approach: non baseline ~
 7 Marine aquatic ecotoxicity (MAETP500) | Problem oriented approach: non basel~
 8 eutrophication (fate not incl.) | Problem oriented approach: baseline (CML, ~
 9 eutrophication (incl. fate, average Europe total, A&B) | Problem oriented ap~
10 Water Consumption Blue - Agriculture
# i 114 more rows
```

```r
exio |>
  filter(year == 2022, region == "US") |>
  group_by(sector) |>
  summarize(total_co2 = sum(.data$value, na.rm = TRUE), .groups = "drop") |>
  slice_max(order_by = total_co2, n = 5) |>
  collect()
```

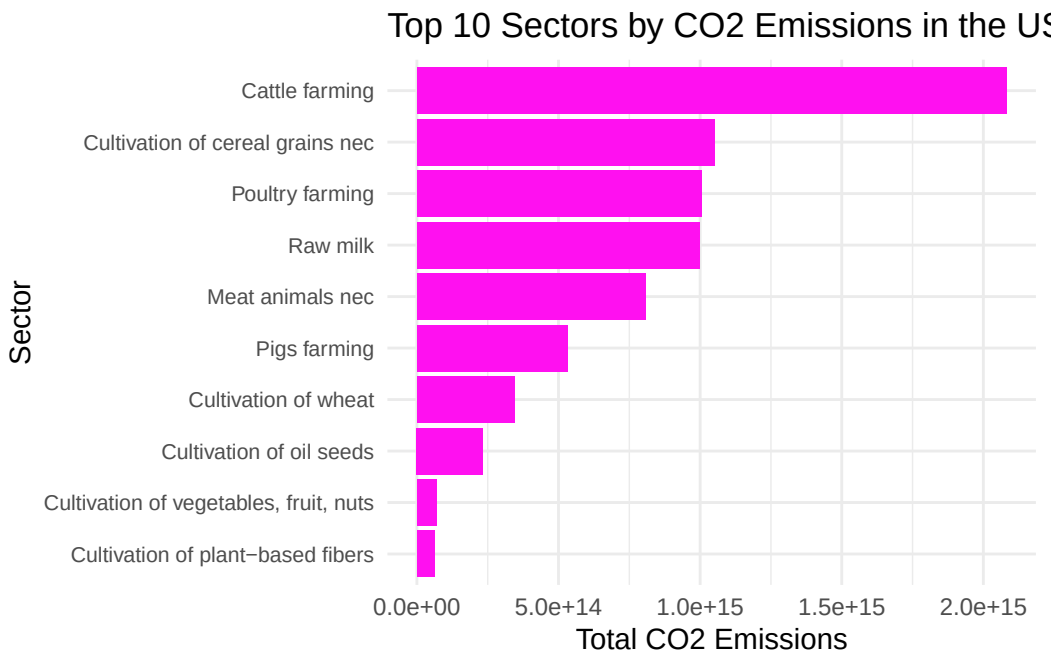
```
# A tibble: 5 x 2
  sector                          total_co2
  <chr>                               <dbl>
1 Cattle farming                    2.08e15
2 Cultivation of cereal grains nec  1.05e15
3 Poultry farming                   1.00e15
4 Raw milk                          9.96e14
5 Meat animals nec                  8.07e14
```

Visualization: Top 10 Sectors by CO2 Emissions

```r
# Get top 10 sectors by CO2 emissions
top10_sectors <- exio |>
  filter(year == 2022, region == "US") |>
  group_by(sector) |>
  summarize(total_co2 = sum(.data$value, na.rm = TRUE), .groups = "drop") |>
  slice_max(order_by = total_co2, n = 10) |>
  collect()

# Create the visualization
p <- ggplot(top10_sectors, aes(x = reorder(sector, total_co2), y = total_co2)) +
  geom_col(fill = "#FF10F0") +
  coord_flip() +
  labs(
    title = "Top 10 Sectors by CO2 Emissions in the US (2022)",
    x = "Sector",
    y = "Total CO2 Emissions"
  ) +
  theme_minimal() +
  theme(axis.text.y = element_text(size = 8))

# Display the plot
print(p)
```




```
# Export the visualization
ggsave("top10_sectors_co2.png", plot = p, width = 10, height = 6, dpi = 300)
```

next task: explore data, which sectors/countries are experiencing decreasing emissions? why is the data weird in the first year?