

Module 1: Tabular Data

Working with larger-than-RAM data using duckdbfs

Authors: Team Madi-Savannah + Bruno

2026-02-18

Table of Contents

1	Introduction	1
2	Case Study: Global Supply Chains	1
3	Setup	2
4	Exercise 1: connecting to remote data	2
5	Exercise 2: Efficient Filtering	2
6	Exercise 3: CO2 Emissions Over Time by Region (Top 5 Emitters)	3
7	Exercise 4: Regions with Largest CO2 Emission Reductions	5
8	Exercise 5: CO2 Emissions Over Time for Top 5 Emitting Countries	9

1 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.

2 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.

3 Setup

Warning: package 'duckdbfs' was built under R version 4.5.2

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

4 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.

```
[1] 1
Rows: ??
Columns: 8
Database: DuckDB 1.4.3 [madis@Windows 10 x64:R 4.5.1/: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~
```

5 Exercise 2: Efficient Filtering

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

```
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~
```

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.

```
# Solution: Top 5 sectors in US by CO2 emissions (2022)

# Target the F_satellite matrix directly to access stressor column
exio_f_sat <- open_dataset(
  "s3://us-west-2.opendata.source.coop/youssef-harby/exiobase-3/4588235/parquet/year=2022/"
)

exio_f_sat |>
  filter(
    grepl("CO2", stressor)
  ) |>
  group_by(sector) |>
  summarize(total_co2 = sum(value, na.rm = TRUE), .groups = "drop") |>
  arrange(desc(total_co2)) |>
  head(5) |>
  collect()

# A tibble: 5 x 2
  sector                                total_co2
  <chr>                                <dbl>
1 Production of electricity by coal      8.33e12
2 Manufacture of cement, lime and plaster 2.54e12
3 Production of electricity by gas       2.30e12
4 Steam and hot water supply            2.07e12
5 Manufacture of basic iron and steel and of ferro-alloys and first p~ 1.84e12
```

6 Exercise 3: CO2 Emissions Over Time by Region (Top 5 Emitters)

First, let's identify the top 5 CO2 emitting regions in 2022:

```
# Find the top 5 emitting regions in 2022
top_emitters_2022 <- exio |>
  filter(year == 2022,
    matrix == "F_satellite",
    stressor %like% "%CO2%") |>
  group_by(region) |>
  summarise(total_co2 = sum(value, na.rm = TRUE)) |>
  arrange(desc(total_co2)) |>
  head(5) |>
  collect()
```

```
top_emitters_2022
```

```
# A tibble: 5 x 2
  region total_co2
  <chr>      <dbl>
1 CN        2.24e13
2 US        7.43e12
3 IN        5.31e12
4 WA        4.12e12
5 WM        4.11e12
```

Now, let's get the time series data for these top 5 regions across all years:

`summarise()` has grouped output by "year". You can override using the `groups` argument.

```
# A tibble: 20 x 3
# Groups:   year [4]
  year region total_co2
  <dbl> <chr>      <dbl>
1  1995 CN        1.43e13
2  1995 IN        3.70e12
3  1995 US        7.50e12
4  1995 WA        2.74e12
5  1995 WM        2.73e12
6  1996 CN        5.74e12
7  1996 IN        1.36e12
8  1996 US        7.56e12
9  1996 WA        1.71e12
10 1996 WM        1.50e12
11 1997 CN        5.81e12
12 1997 IN        1.43e12
13 1997 US        7.76e12
14 1997 WA        1.71e12
15 1997 WM        1.57e12
16 1998 CN        5.96e12
17 1998 IN        1.47e12
18 1998 US        7.95e12
19 1998 WA        1.70e12
20 1998 WM        1.60e12
```

Create a line plot showing CO2 emissions over time for the top 5 regions:

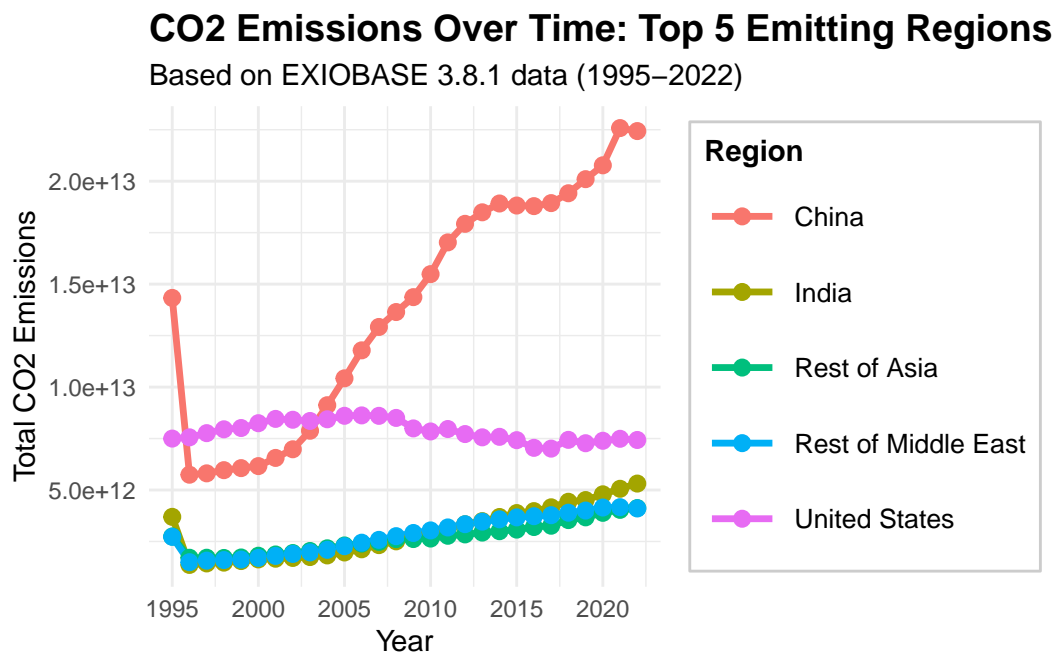
```
# Create the time series plot
co2_plot <- ggplot(co2_timeseries, aes(x = year, y = total_co2, color = region_name)) +
  geom_line(linewidth = 1.2) +
  geom_point(size = 2.5) +
  labs(
```

```

title = "CO2 Emissions Over Time: Top 5 Emitting Regions",
subtitle = "Based on EXIOBASE 3.8.1 data (1995-2022)",
x = "Year",
y = "Total CO2 Emissions",
color = "Region"
) +
theme_minimal() +
theme(
  plot.title = element_text(face = "bold", size = 14),
  legend.position = "right",
  legend.title = element_text(face = "bold", size = 11),
  legend.text = element_text(size = 10),
  legend.background = element_rect(fill = "white", color = "gray80"),
  legend.key.size = unit(1, "cm")
) +
scale_x_continuous(breaks = seq(1995, 2022, by = 5))

```

co2_plot



7 Exercise 4: Regions with Largest CO2 Emission Reductions

Let's identify which regions have reduced their CO2 emissions the most between 1995 and 2022:

```

# Get CO2 emissions for all regions in 1995 and 2022
co2_1995 <- exio |>
  filter(year == 1995,

```

```

    matrix == "F_satellite",
    stressor %like% "%CO2%") |>
group_by(region) |>
summarise(co2_1995 = sum(value, na.rm = TRUE)) |>
collect()

co2_2022 <- exio |>
  filter(year == 2022,
    matrix == "F_satellite",
    stressor %like% "%CO2%") |>
group_by(region) |>
summarise(co2_2022 = sum(value, na.rm = TRUE)) |>
collect()

# Join the two datasets and calculate change
co2_change <- co2_1995 |>
  inner_join(co2_2022, by = "region") |>
  mutate(
    absolute_change = co2_2022 - co2_1995,
    percent_change = ((co2_2022 - co2_1995) / co2_1995) * 100
  ) |>
  filter(grepl("^W", region)) |> # Include only rest-of-world regions (WA, WE, WF, WL, WM)
  arrange(absolute_change)

# View regions with the largest absolute decreases
head(co2_change, 5)

```

```

# A tibble: 5 x 5
  region co2_1995 co2_2022 absolute_change percent_change
<chr>    <dbl>    <dbl>         <dbl>         <dbl>
1 WE      9.03e11  6.71e11      -2.32e11       -25.7
2 WL     1.03e12  1.21e12       1.78e11        17.3
3 WF      6.19e11  1.05e12       4.26e11        68.7
4 WM      2.73e12  4.11e12       1.38e12        50.6
5 WA      2.74e12  4.12e12       1.38e12        50.4

```

Now let's visualize the top 5 regions with the largest absolute reductions:

```

# Get top 5 reducers
top_reducers <- co2_change |>
  head(5)

# Create a bar plot
reduction_plot <- ggplot(top_reducers, aes(x = reorder(region, absolute_change), y = absolute_change)) +
  geom_col() +
  coord_flip() +
  scale_fill_brewer(palette = "Set2") +

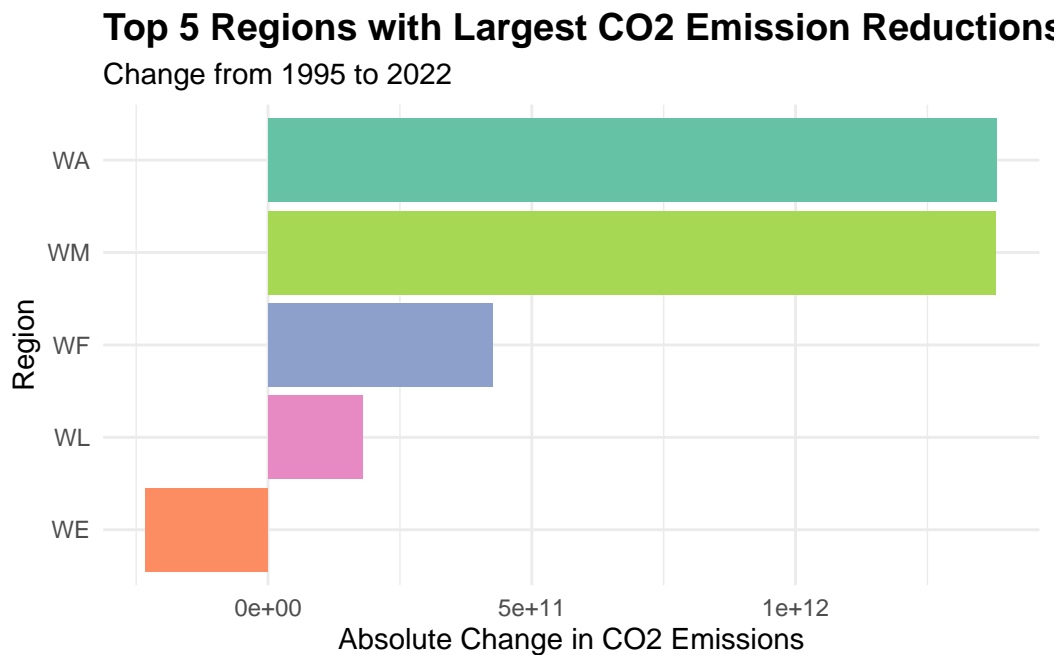
```

```

labs(
  title = "Top 5 Regions with Largest CO2 Emission Reductions",
  subtitle = "Change from 1995 to 2022",
  x = "Region",
  y = "Absolute Change in CO2 Emissions"
) +
theme_minimal() +
theme(
  plot.title = element_text(face = "bold", size = 14),
  legend.position = "none"
)

reduction_plot

```



Let's also look at percentage reductions (excluding countries with very low baseline emissions):

```

# Filter to countries with substantial 1995 emissions (>1000 units) to avoid outliers
substantial_emitters <- co2_change |>
  filter(co2_1995 > 1000) |>
  arrange(percent_change) |>
  head(5)

substantial_emitters

```

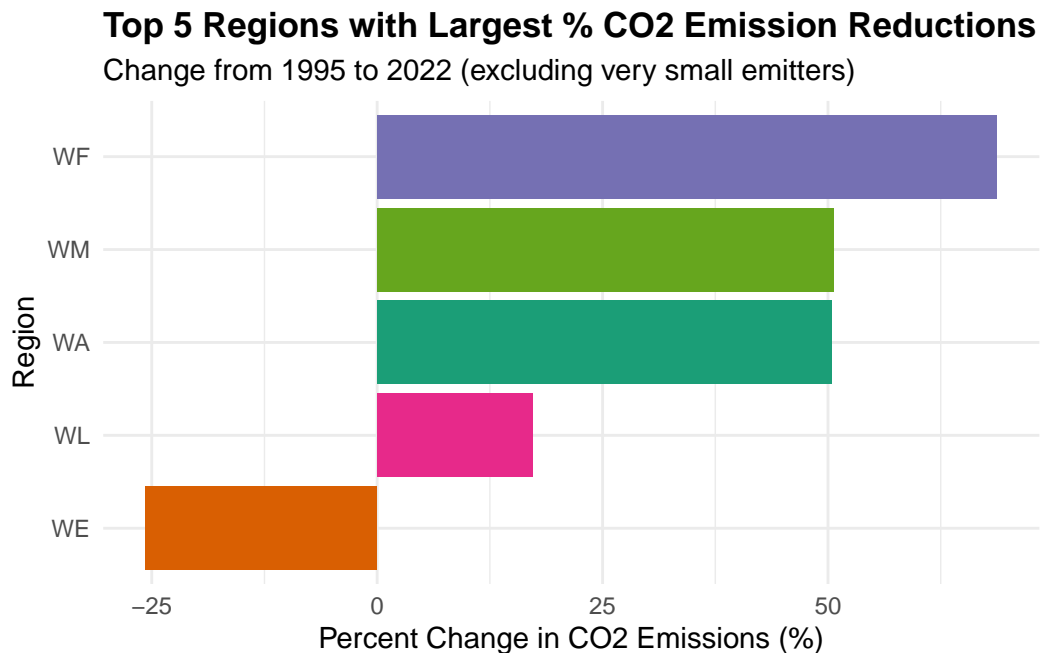
A tibble: 5 x 5

	region	co2_1995	co2_2022	absolute_change	percent_change
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	WE	9.03e11	6.71e11	-2.32e11	-25.7

2	WL	1.03e12	1.21e12	1.78e11	17.3
3	WA	2.74e12	4.12e12	1.38e12	50.4
4	WM	2.73e12	4.11e12	1.38e12	50.6
5	WF	6.19e11	1.05e12	4.26e11	68.7

```
# Visualize percentage reductions
percent_plot <- ggplot(substantial_emitters, aes(x = reorder(region, percent_change), y = pe
  geom_col() +
  coord_flip() +
  scale_fill_brewer(palette = "Dark2") +
  labs(
    title = "Top 5 Regions with Largest % CO2 Emission Reductions",
    subtitle = "Change from 1995 to 2022 (excluding very small emitters)",
    x = "Region",
    y = "Percent Change in CO2 Emissions (%)"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 13),
    legend.position = "none"
  )
)

percent_plot
```



8 Exercise 5: CO2 Emissions Over Time for Top 5 Emitting Countries

Now let's look at individual countries (excluding rest-of-world aggregates) to see the top 5 emitting countries:

```
# Find the top 5 emitting countries in 2022 (excluding W regions)
top_countries_2022 <- exio |>
  filter(year == 2022,
         matrix == "F_satellite",
         stressor %like% "%CO2%") |>
  group_by(region) |>
  summarise(total_co2 = sum(value, na.rm = TRUE)) |>
  filter(!grepl("^W", region)) |> # Exclude rest-of-world regions
  arrange(desc(total_co2)) |>
  head(5) |>
  collect()
```

```
top_countries_2022
```

```
# A tibble: 5 x 2
  region total_co2
  <chr>      <dbl>
1 CN        2.24e13
2 US        7.43e12
3 IN        5.31e12
4 RU        3.12e12
5 JP        2.01e12
```

Get time series data for these top 5 countries across all years:

`summarise()` has grouped output by "year". You can override using the `.groups` argument.

```
# A tibble: 20 x 3
# Groups:   year [4]
  year region total_co2
  <dbl> <chr>      <dbl>
1 1995 CN        1.43e13
2 1995 IN        3.70e12
3 1995 JP        2.21e12
4 1995 RU        2.90e12
5 1995 US        7.50e12
6 1996 CN        5.74e12
7 1996 IN        1.36e12
8 1996 JP        2.06e12
9 1996 RU        2.79e12
10 1996 US        7.56e12
11 1997 CN        5.81e12
```

```

12 1997 IN      1.43e12
13 1997 JP      2.06e12
14 1997 RU      2.66e12
15 1997 US      7.76e12
16 1998 CN      5.96e12
17 1998 IN      1.47e12
18 1998 JP      2.02e12
19 1998 RU      2.54e12
20 1998 US      7.95e12

```

Create a line plot showing CO2 emissions over time for the top 5 countries:

```

# Create the time series plot
countries_plot <- ggplot(co2_countries_timeseries, aes(x = year, y = total_co2, color = country)) +
  geom_line(linewidth = 1.2) +
  geom_point(size = 2.5) +
  labs(
    title = "CO2 Emissions Over Time: Top 5 Emitting Countries",
    subtitle = "Based on EXIOBASE 3.8.1 data (1995-2022)",
    x = "Year",
    y = "Total CO2 Emissions",
    color = "Country"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 14),
    legend.position = "right",
    legend.title = element_text(face = "bold", size = 11),
    legend.text = element_text(size = 10),
    legend.background = element_rect(fill = "white", color = "gray80"),
    legend.key.size = unit(1, "cm")
  ) +
  scale_x_continuous(breaks = seq(1995, 2022, by = 5))

countries_plot

```

