Fetching and Visualizing Official Statistics with R

Nicolas
Eesti Pank

2025-04-17

Table of contents

Interfaces to Official Statistics

- Packages or set of classes and methods to read data and metadata documents through exchange frameworks
  - Use R (or Python) packages to read data from APIs, databases, and web pages
    - Individual packages:
      - eurostat: Access data from Eurostat
      - OECD: Access data from the OECD API
  - General-purpose packages:
    - rdbnomics: Unified access to many economic databases (e.g. ECB, **Eurostat**, IMF, World Bank)

- Interface standards:
  - SDMX: Statistical Data and Metadata Exchange format
  - pxweb: Access to data sources using the PX-Web API (e.g. Statistics Sweden, Statistics Estonia)

DBnomics

- DBnomics is a database of databases
  - free platform to aggregate publicly-available economic data provided by national and international statistical institutions, but also by researchers and private companies
  - Unified interface to access data from many sources
  - Harmonized data formats and metadata
  - Data series are available upon release by the provider
  - Each revision is archived to build a real-time database

How to fetch data (from DBnomics using R)

- DBnomics R client

```r
install.packages("rdbnomics")
library(rdbnomics)
```

Packages used in this tutorial

- 📦 Fetching data (`rdbnomics`)
- 🧹 Data wrangling and transformation (`tidyverse`)
- 📊 Visualization (`ggplot2`, `plotly`)
- 📋 Tabular summaries (`gt`)
- 🧾 Building this presentation (`quarto`)

```r
library(quarto)     # for compiling Quarto presentations
library(rdbnomics)  # for accessing economic data via DBnomics
library(tidyverse)  # dplyr, ggplot2, readr, etc.
library(plotly)     # interactive visualizations
library(gt)         # pretty tables
```

Example: Fetch Unemployment Data

- Assume we know exactly the series ID we want to fetch
  - Unemployment rate, ILO definition, total, Estonia, from Eurostat

```
unemp <- rdb(ids = "Eurostat/ei_lmhr_m/M.PC_ACT.SA.LM-UN-T-TOT.EE") # fetch data
```

```r
glimpse(unemp)
```

```
Rows: 296
Columns: 22
$ `@frequency`                       <chr> "monthly", "monthly", "monthly", "mo…
$ dataset_code                       <chr> "ei_lmhr_m", "ei_lmhr_m", "ei_lmhr_m…
$ dataset_name                       <chr> "Unemployment rate (%) - monthly dat…
$ freq                               <chr> "M", "M", "M", "M", "M", "M", "M", "…
$ geo                                <chr> "EE", "EE", "EE", "EE", "EE", "EE", …
$ `Geopolitical entity (reporting)` <chr> "Estonia", "Estonia", "Estonia", "Es…
$ indexed_at                         <dttm> 2024-10-31 15:26:51, 2024-10-31 15:…
$ indic                              <chr> "LM-UN-T-TOT", "LM-UN-T-TOT", "LM-UN…
$ Indicator                          <chr> "Unemployment according to ILO defin…
$ observations_attributes            <chr> "OBS_FLAG,", "OBS_FLAG,", "OBS_FLAG,…
$ original_period                    <chr> "2000-02", "2000-03", "2000-04", "20…
$ original_value                     <chr> "14.9", "14.2", "14.5", "13.9", "14"…
$ period                             <date> 2000-02-01, 2000-03-01, 2000-04-01,…
$ provider_code                      <chr> "Eurostat", "Eurostat", "Eurostat", …
$ s_adj                              <chr> "SA", "SA", "SA", "SA", "SA", "SA", …
$ `Seasonal adjustment`              <chr> "Seasonally adjusted data, not calen…
$ series_code                        <chr> "M.PC_ACT.SA.LM-UN-T-TOT.EE", "M.PC_…
$ series_name                        <chr> "Monthly - Percentage of population …
```

```r
colnames(unemp)
```

```
 [1] "@frequency"                      "dataset_code"
 [3] "dataset_name"                    "freq"
 [5] "geo"                             "Geopolitical entity (reporting)"
 [7] "indexed_at"                      "indic"
 [9] "Indicator"                       "observations_attributes"
[11] "original_period"                 "original_value"
[13] "period"                          "provider_code"
[15] "s_adj"                           "Seasonal adjustment"
[17] "series_code"                     "series_name"
[19] "Time frequency"                  "unit"
[21] "Unit of measure"                 "value"
```

```r
# Extract source and series ID from the metadata
(source_name <- unique(unemp$dataset_code))
```

```
[1] "ei_lmhr_m"
```

```r
(provider_code <- unique(unemp$provider_code))
```

```
[1] "Eurostat"
```

```r
(country_name <- unique(unemp$`Geopolitical entity (reporting)`)  )
```

```
[1] "Estonia"
```

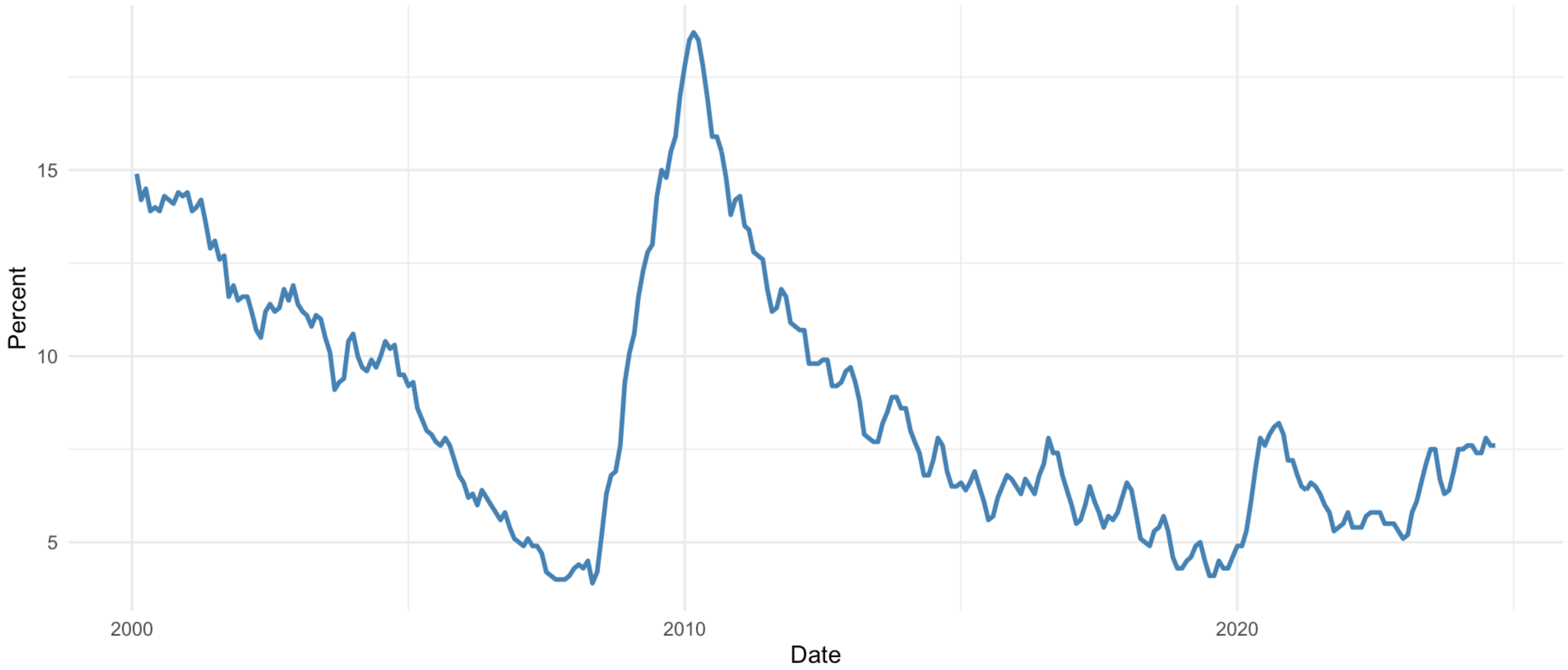
```r
(series_id <- unique(unemp$series_code))
```

```
[1] "M.PC_ACT.SA.LM-UN-T-TOT.EE"
```

```r
# Plot the data
p1 <- ggplot(unemp, aes(x = period, y = value)) +
  geom_line(color = "steelblue", linewidth = 1) +
  labs(
    title = paste("Unemployment Rate in ", country_name),
    subtitle = paste("Monthly, seasonally adjusted -", provider_code),
    x = "Date", y = "Percent",
    caption = paste("Source:", provider_code, "| Dataset:", source_name, "| ID:", series_id)
  ) +
  theme_minimal()
p1
```

Unemployment Rate in Estonia

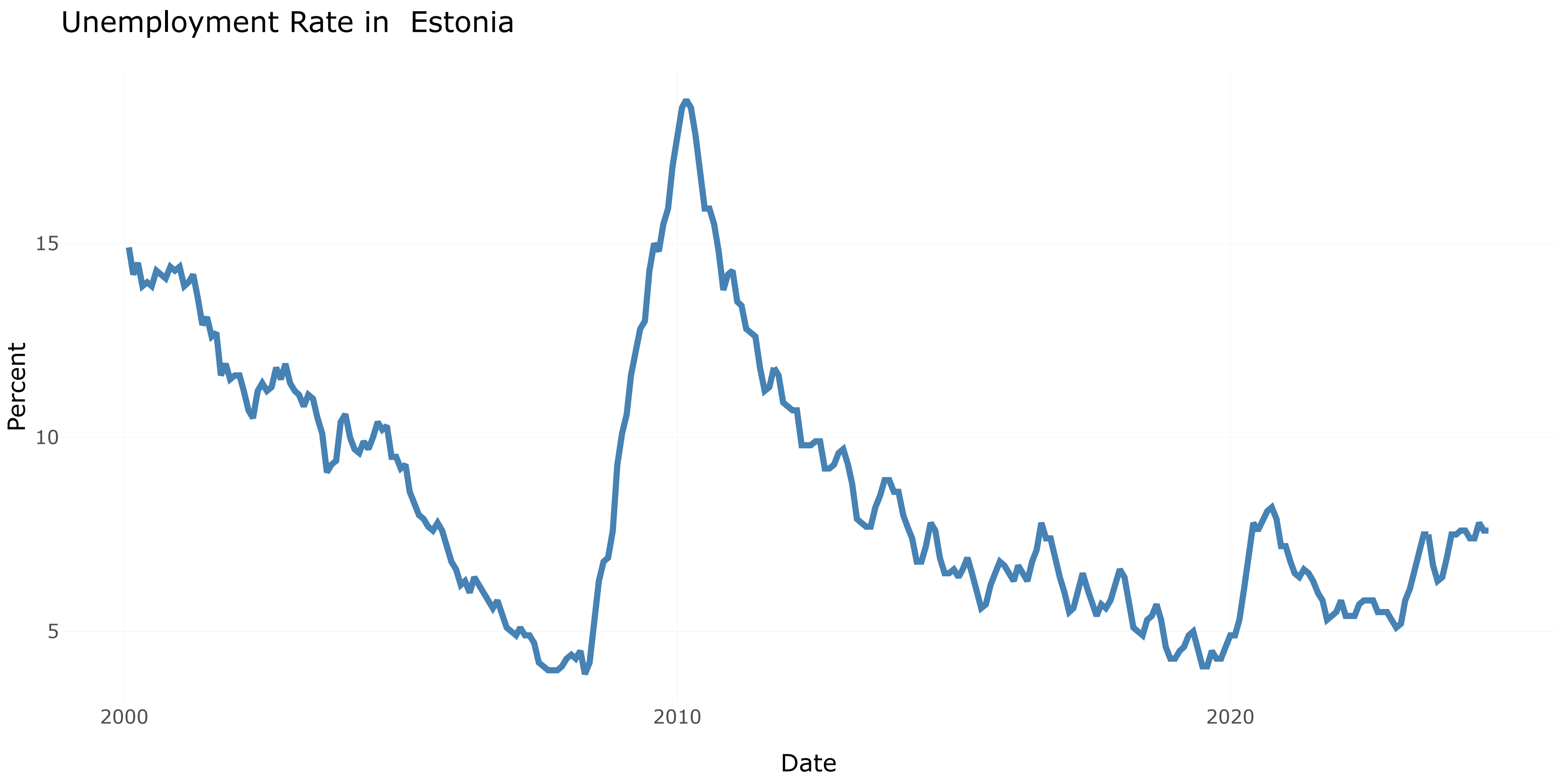
Monthly, seasonally adjusted — Eurostat



Source: Eurostat | Dataset: ei_lmhr_m | ID: M.PC_ACT.SA.LM-UN-T-TOT.EE

Interactive plot

```r
ggplotly(p1)
```



How do we find the series ID/mask/dimensions?

- Go to the DBnomics website
    - Search directly for a series or pick a provider
    - Search for the data you want (dataset_code)
    - Click on the series (series_code)
    - Copy the series ID from the URL

- Show the available datasets of a provider:

```r
head(rdb_datasets(provider_code = "Eurostat"))
```

```
$Eurostat
                 code
               <char>
   1:    aact_ali01
   2:    aact_ali02
   3:    aact_eaa01
   4:    aact_eaa02
   5:    aact_eaa03
  ---
8289: yth_empl_120
8290: yth_empl_130
8291: yth_empl_130
8292: yth_empl_140
8293: yth_empl_140
                                                                              name
                                                                            <char>
   1:       Agricultural labour input statistics: absolute figures (1 000 annual work units)
   2:                                                Agricultural labour input statistics: indices
   3:                              Economic accounts for agriculture - values at current prices
   4:                                  Economic accounts for agriculture - values at n-1 prices
```

- Show the dimensions of a dataset:

```r
head(rdb_dimensions(provider_code = "Eurostat", dataset_code = "ei_lmhr_m"))
```

```
$Eurostat
$Eurostat$ei_lmhr_m
$Eurostat$ei_lmhr_m$freq
     freq Time frequency
   <char>         <char>
1:      M        Monthly

$Eurostat$ei_lmhr_m$geo
          geo          Geopolitical entity (reporting)
       <char>                                   <char>
 1:        AT                                  Austria
 2:        BA                   Bosnia and Herzegovina
 3:        BE                                  Belgium
 4:        BG                                 Bulgaria
 5:        CH                              Switzerland
 6:        CY                                   Cyprus
 7:        CZ                                  Czechia
 8:        DE                                  Germany
 9:        DK                                  Denmark
10:      EA20      Euro area - 20 countries (from 2023)
```

- Query to filter/select series from a provider’s dataset

```
head(rdb_series(
  provider = "Eurostat",
  dataset_code = "ei_lmhr_m",
  query = "United Kingdom"
))
```

```
$Eurostat
$Eurostat$ei_lmhr_m
                          series_code
                               <char>
 1: M.PC_ACT.NSA.LM-UN-F-GT25.UK
 2: M.PC_ACT.NSA.LM-UN-F-LE25.UK
 3:  M.PC_ACT.NSA.LM-UN-F-TOT.UK
 4: M.PC_ACT.NSA.LM-UN-M-GT25.UK
 5: M.PC_ACT.NSA.LM-UN-M-LE25.UK
 6:  M.PC_ACT.NSA.LM-UN-M-TOT.UK
 7: M.PC_ACT.NSA.LM-UN-T-GT25.UK
 8: M.PC_ACT.NSA.LM-UN-T-LE25.UK
 9:  M.PC_ACT.NSA.LM-UN-T-TOT.UK
10:  M.PC_ACT.SA.LM-UN-F-GT25.UK
11:  M.PC_ACT.SA.LM-UN-F-LE25.UK
12:   M.PC_ACT.SA.LM-UN-F-TOT.UK
13:  M.PC_ACT.SA.LM-UN-M-GT25.UK
14:  M.PC_ACT.SA.LM-UN-M-LE25.UK
15:   M.PC_ACT.SA.LM-UN-M-TOT.UK
16:  M.PC_ACT.SA.LM-UN-T-GT25.UK
```

Fetch two (or more) series at once

- Example: Balance of Payments (BOP) for France and Germany from the IMF for Current Account, Total, Net, Euros, Millions, Annual

Option A   Option B: Option C:

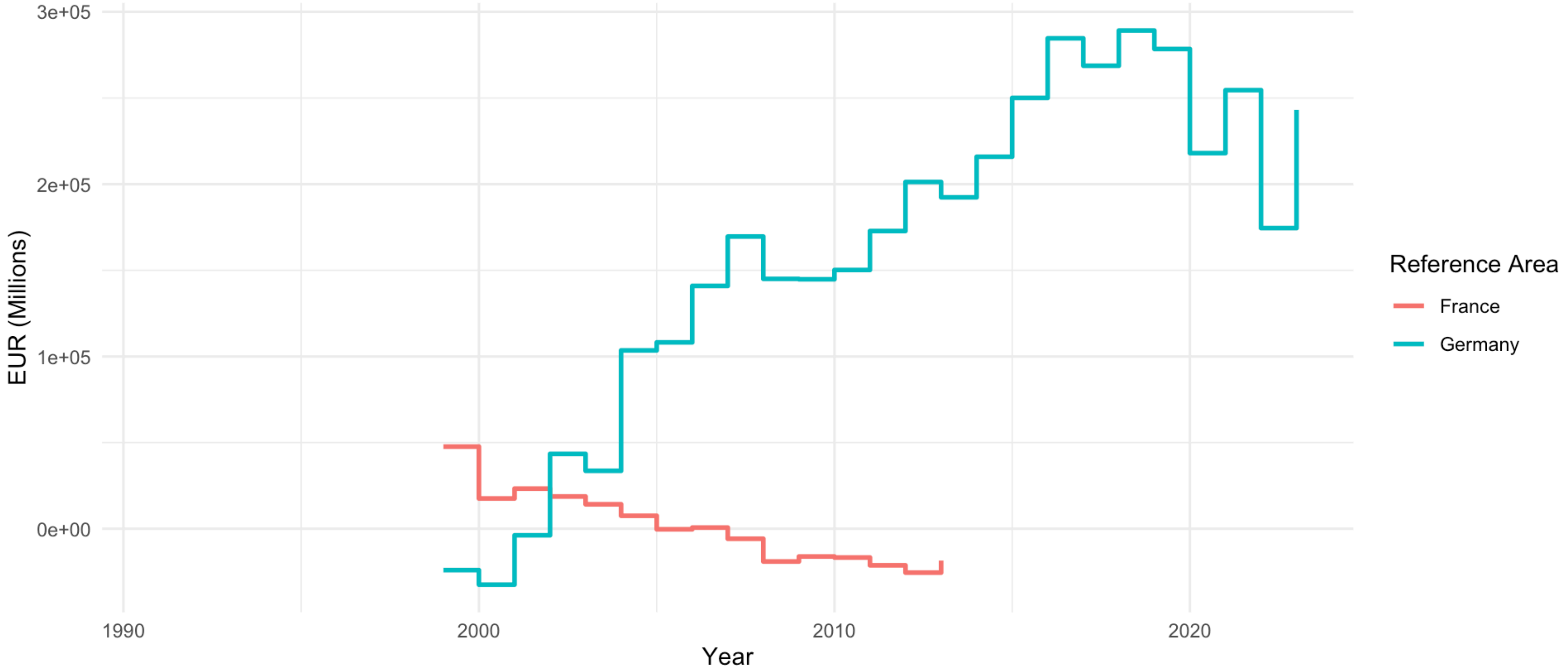
```r
# by ID
bop <- rdb(ids = c("IMF/BOP/A.FR.BCA_BP6_EUR", "IMF/BOP/A.DE.BCA_BP6_EUR"))
bop %>% count(`Reference Area`)
```

```
   Reference Area     n
           <char> <int>
1:         France    15
2:        Germany    26
```

```r
# Line plot with color by country
p2 <- ggplot(bop, aes(x = period, y = value, color = `Reference Area`)) +
  geom_step(linewidth = 1) +
  labs(
    title = "Balance of Payments (BCA, EUR)",
    subtitle = "France vs Germany – Annual",
    x = "Year",
    y = "EUR (Millions)",
    caption = "Source: IMF / DBnomics"
  ) +
  theme_minimal()
p2
```

Balance of Payments (BCA, EUR)

France vs Germany — Annual



Source: IMF / DBnomics

Fetch two series from different datasets of different providers

```r
unemp2 <- rdb(ids = c("AMECO/ZUTN/EA19.1.0.0.0.ZUTN", "Eurostat/une_rt_q/Q.SA.Y15-24.PC_ACT.T.EA19"))
```

```r
# See which providers and datasets are included
dim(unemp2)
```

```
[1] 122  27
```

```r
unique(unemp2$provider_code)
```

```
[1] "AMECO"    "Eurostat"
```

```r
unique(unemp2$dataset_code)
```

```
[1] "ZUTN"     "une_rt_q"
```

```r
unique(unemp2$series_code)
```

```
[1] "EA19.1.0.0.0.ZUTN"         "Q.SA.Y15-24.PC_ACT.T.EA19"
```

```r
unique(unemp2$`@frequency`)
```

```
[1] "annual"    "quarterly"
```

```r
unique(unemp2$`Seasonal adjustment`)
```

```
[1] NA
[2] "Seasonally adjusted data, not calendar adjusted data"
```

```r
# Summarize coverage and data availability
unemp2_summary <- unemp2 %>%
  group_by(series_code) %>%
  summarize(
    provider = first(provider_code),
    dataset = first(dataset_code),
    start_all = min(period, na.rm = TRUE),
    end_all = max(period, na.rm = TRUE),
    start_data = min(period[!is.na(value)]),
    end_data = max(period[!is.na(value)]),
    n_obs = sum(!is.na(value)),
    .groups = "drop"
  )
```

```r
unemp2_summary_table <- unemp2_summary |>
gt() %>%
  tab_header(
    title = "Time Coverage and Non-Missing Observations",
    subtitle = "For Each Series from AMECO and Eurostat"
  ) %>%
  cols_label(
    series_code = "Series ID",
    provider = "Provider",
    dataset = "Dataset",
    start_all = "Start (all)",
    end_all = "End (all)",
    start_data = "Start (non-NA)",
    end_data = "End (non-NA)",
    n_obs = "# Obs"
  ) %>%
  fmt_date(
    columns = c(start_all, end_all, start_data, end_data),
    date_style = "iso"
  ) %>%
  tab_options(
    table.width = pct(100),
    column_labels.font.weight = "bold"
  )
```

```r
unemp2_summary_table
```

Time Coverage and Non-Missing Observations

For Each Series from AMECO and Eurostat

Series ID	Provider	Dataset	Start (all)	End (all)	Start (non-NA)	End (non-NA)	# Obs
EA19.1.0.0.0.ZUTN	AMECO	ZUTN	1960-01-01	2026-01-01	1997-01-01	2026-01-01	30
Q.SA.Y15-24.PC_ACT.T.EA19	Eurostat	une_rt_q	2009-01-01	2022-07-01	2009-01-01	2022-07-01	55

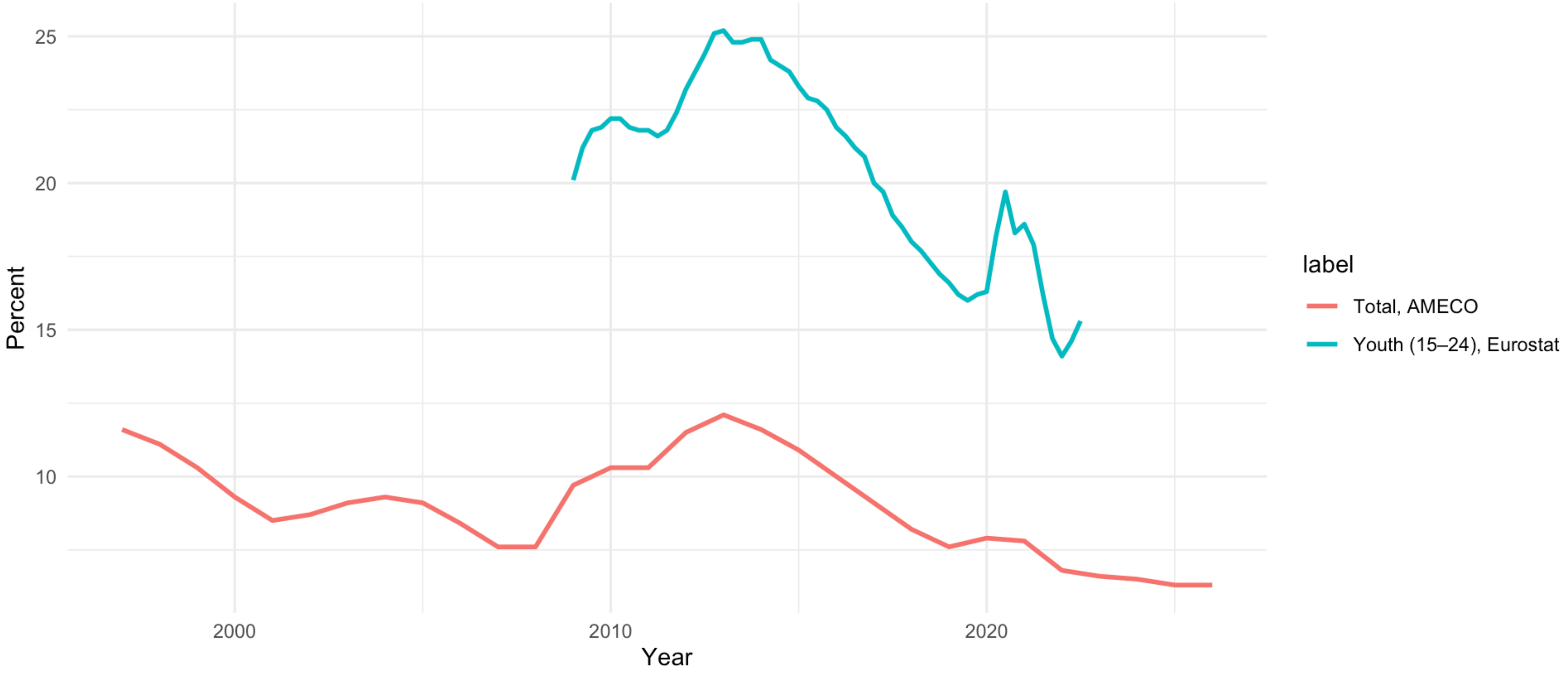
```r
# Metadata vectors
providers <- unique(unemp2$provider_code)
datasets <- unique(unemp2$dataset_code)
series_ids <- unique(unemp2$series_code)
```

```r
# Create a label that combines dataset + series ID
unemp2_clean <- unemp2 %>%
  drop_na(value) %>%
  mutate(label = case_when(
    series_code == "EA19.1.0.0.0.ZUTN" ~ "Total, AMECO",
    series_code == "Q.SA.Y15-24.PC_ACT.T.EA19" ~ "Youth (15–24), Eurostat",
    TRUE ~ series_code
  ))
```

```r
p3 <- ggplot(unemp2_clean, aes(x = period, y = value, color = label)) +
  geom_line(linewidth = 1) +
  labs(
    title = "Unemployment Rates from Multiple Sources (EA19)",
    subtitle = "AMECO and Eurostat - Different definitions",
    x = "Year", y = "Percent",
    caption = paste("Series IDs:", paste(unique(unemp2_clean$series_code), collapse = " | "))
  ) +
  theme_minimal()
p3
```

Unemployment Rates from Multiple Sources (EA19)

AMECO and Eurostat — Different definitions



Series IDs: EA19.1.0.0.0.ZUTN | Q.SA.Y15-24.PC_ACT.T.EA19

Fetch large amounts of data

- Sometimes you need to fetch many if not all dimensions of the data
- You can wildcard dimension and post-filter
- Example: MFI Interest Rate Statistics from the ECB
    - Start with a single series (Estonia, mortgage rates)

```r
mir_mortgage_ee <- rdb("ECB", "MIR", "M.EE.B.A2C.A.R.A.2250.EUR.N")
unique(mir_mortgage_ee$series_name)
```

```
[1] "Monthly - Estonia - Deposit-taking corporations except the central bank (S.122) - Lending for house
purchase excluding revolving loans and overdrafts, convenience and extended credit card debt - Total -
Annualised agreed rate (AAR) / Narrowly defined effective rate (NDER) - Total - Households and non-profit
institutions serving households (S.14 and S.15) - Euro - New business"
```

Wildcarding dimensions

- To fetch **multiple values** for a dimension (e.g. countries), just **remove** the value from that position
    - Example: remove "EE" to fetch all countries (REF_AREA)

⚠ This can take a while

```r
# mir_mortgage_ee <- rdb("ECB", "MIR", "M.EE.B.A2C.A.R.A.2250.EUR.N")
mir <- rdb("ECB", "MIR", "M..B..A.R.A..EUR.N")
unique(mir$REF_AREA)
```

```
 [1] "AT" "BE" "CY" "DE" "EE" "ES" "FI" "FR" "GR" "HR" "IE" "IT" "LT" "LU" "LV"
[16] "MT" "NL" "PT" "SI" "SK" "U2"
```

```r
unique(mir$BS_ITEM)
```

```
 [1] "A2A"  "A2AC" "A2B"  "A2BC" "A2C"  "A2CC" "A2D"  "A2Z"  "A2Z1" "A2Z3"
[11] "L21"  "L22"  "L23"  "L24"
```

```r
unique(mir$`BS counterpart sector`)
```

```
[1] "Non-Financial corporations (S.11)"
[2] "Households and non-profit institutions serving households (S.14 and S.15)"
[3] "Households of which sole proprietors and unincorporated partnerships (SP/UP)"
[4] "Non-Financial corporations and Households (S.11 and S.14 and S.15)"
```

Filter and plot

- Filter Estonia, Latvia and Lithuania
- Keep only selected **BS items** (loan categories)

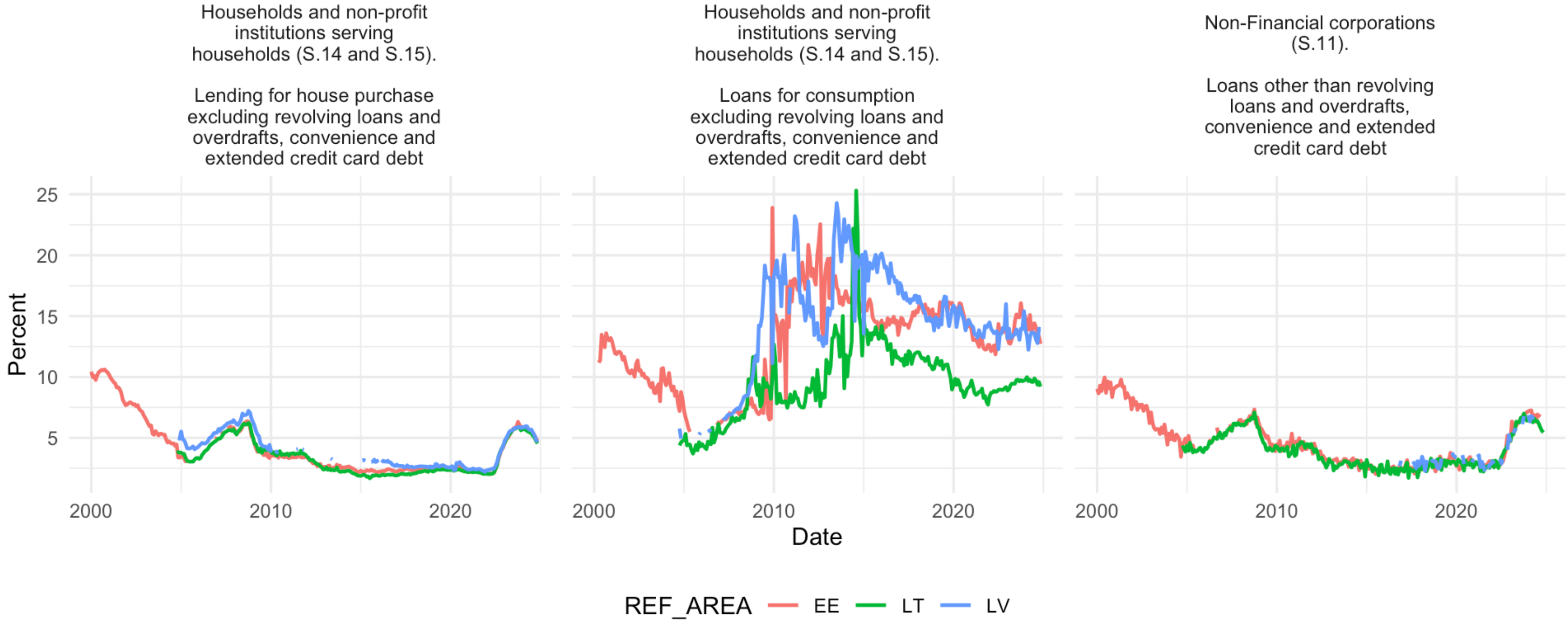
```r
# Filter by BS_ITEM and countries
mir_filtered <- mir %>%
  filter(
    REF_AREA %in% c("EE", "LV", "LT"),
    BS_ITEM %in% c("A2I", "A2C", "A2B", "A2J", "A2A")
  )
```

Plot interest rates by country & type

```r
country_list <- paste(sort(unique(mir_filtered$REF_AREA)), collapse = ", ")
item_list <- paste(unique(mir_filtered$BS_ITEM), collapse = ", ")

caption_text <- paste(
  "Source: ECB / DBnomics – Dataset code: MIR",
  paste0("\nFiltered: REF_AREA in ", country_list, "; BS_ITEM in ", item_list)
)

mir_filtered <- mir_filtered %>%
  mutate(facet_label = paste0(`BS counterpart sector`, ".\n\n", `Balance sheet item`))

p4 <- ggplot(mir_filtered, aes(x = period, y = value, color = REF_AREA)) +
  geom_line(linewidth = 0.8) +
  facet_wrap(~ facet_label, labeller = label_wrap_gen(width = 30), ncol = 3) +
  labs(
    title = "Interest Rates for Households and Firms",
    subtitle = "Faceted by Loan Type and Borrower Sector",
    x = "Date", y = "Percent",
    caption = caption_text
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
p4
```

Interest Rates for Households and Firms

Faceted by Loan Type and Borrower Sector



Source: ECB / DBnomics — Dataset code: MIR
Filtered: REF_AREA in EE, LT, LV; BS_ITEM in A2A, A2B, A2C