Intersecting Inequalities

Supplementary Materials

Stephanie Steinmetz (UNIL) Mattia Guarnerio (UNIL)

Tuesday, September 16, 2025

Table of contents

1. Introduction

Our book chapter explores how major crises, specifically the **Great Recession** and the **COVID-19 pandemic**, have shaped occupational outcomes in the European Union through an *intersectional* lens, focusing on the compounded disadvantages associated with **gender** and **migrant background**. Emphasising the growing importance of migrant women in the European labour force, it calls for a critical rethinking of conventional explanations of gendered labour market impacts during times of crisis, namely the *buffer*, *substitution*, and *segmentation* hypotheses, to account for the multiple vulnerabilities faced by an increasing heterogeneous workforce.

Drawing on pooled cross-sectional data from the *European Union Statistics on Income and Living Conditions* (EU-SILC) survey for the years 2007 to 2022 across 20 EU countries (Eurostat, 2024), the chapter examines gender- and migration-based inequalities in **employment status**, **job quality**, and **occupational segregation**.

Findings reveal that native workers remained relatively insulated from both crises. Notably, native women no longer fulfil their historical role as a flexible labour reserve. Instead, non-EU migrant workers, especially women, have assumed this buffering role, disproportionately absorbing employment shocks during economic downturns. These insights underscore the need to systematically integrate an intersectional perspective into labour market research and policymaking in order to fully capture how crises differentially affect Europe's increasingly diversifying workforce.

This document illustrates the practical execution of all data tidying steps, taken on an already modified sample of EU-SILC, in order to ensure privacy and anonimity. Furthermore, it details the generation and formatting of all tables and figures contained in the book chapter, both in the main text and the online appendix. The main aim is to guarantee full transparency and replicability of our results and findings.

To get started, the necessary R (R Core Team, 2021) `version 4.3.3` packages must be loaded from the library into the environment. Notable required extensions are:

1. `future` `version 1.40.0` (Bengtsson, 2015), for speeding up the calculation of the **Mutual Information Index of Occupational Segregation** (**M**) through *parallel processing* (see Guinea-Martin et al., 2015).
2. `ggtext` `version 0.1.2` (Wilke & Wiernik, 2020), for rendering text boxes with complex formatting.
3. `gridExtra` `version 2.3` (Auguie, 2010), for arranging multiple grid-based plots in a single figure.
4. `gt` `version 1.0.0` (Iannone et al., 2025), for creating beautifully formatted, LaTeX-ready display tables.
5. `gtExtras` `version 0.5.0` (Mock, 2022), for supporting more complex formatting with `gt`.
6. `haven` `version 2.5.4` (Wickham et al., 2015), for importing and exporting Stata (`.dta`) files.
7. `mutualinf` `version 2.0.3` (Angulo-Gonzalez et al., 2021), for computing **M**.
8. `srvyr` `version 1.3.0` (Freedman Ellis & Schneider, 2016), for applying cross-sectional personal weights.
9. `tidyverse` `version 2.0.0` (Wickham et al., 2019), for data tidying, wrangling, and visualisation, including `ggplot2` `version 3.5.2` (Wickham, 2016).

2. Data Tidying

EU-SILC microdata must be tidied and prepared for tabulation and visualisation, applying sample restrictions and recoding variable labels. The original EU-SILC sources cannot be published in an open source format, due to data privacy requirements related to the scientific use files provided by Eurostat.

Accordingly, we execute the following steps on the main sample:

1. Importing the EU-SILC dataset and removing all superfluous variables.
2. Recoding the *country* variable to standardise all codes.
3. Transforming *sexmig*, *year*, *country*, *isco*, *PL050*, *PL051*, *occrit*, and *emp* into factor variables.
4. Filtering out all self-employed individuals from the main analytic sample, based on *PL040*.
5. Creating an analytic subsample containing only employed respondents, based on *emp*, and removing *temp* and *PL060* from the main sample.
6. Filtering out all individuals who work more than 70 hours per week from the employed subsample, based on *PL060*.
7. Apply casewise deletion to all analytic samples.

2.1 Importing EU-SILC data

EU-SILC data is imported using the `read_dta` function from `haven`.

```
# Importing the "EUSILC_20072022_WF_FIN.dta" Stata file and selecting only
 ↪  the variables that are required for our analysis.

d <- read_dta("Data/EUSILC_20072022_WF_FIN.dta") |>
  select(year, country, sexmig,
         RB050, PL040, emp,
         isco, PL050, PL051,
         occrit, PL060, temp)

# N = 2.442.653 is the original sample size.
```

2.2 Recoding EU-SILC data

In EU-SILC, the *country* variable uses the `EL` and `GR` codes for **Greece**. To ensure consistency and comparability across samples, all values are standardised to `EL`. Moreover, *sexmig*, *year*, *country*, *isco*, *PL050*, *PL051*, *occrit*, and *emp* are transformed into factor variables with easily readable labels.

```
# Setting all codes for Greece are all set to "EL", using case_when() in a
 ↪  mutate() call.
d <- d |> mutate(
  country = case_when(
    country %in% c("GR") ~ "EL",
    TRUE ~ country
  )
) |>
  # Transforming all variables into factors, using as.factor() in a mutate()
   ↪  call.
  mutate(
    sexmig = as.factor(sexmig),
    year = as.factor(year),
    country = as.factor(country),
    occrit = as.factor(occrit),
    emp = as.factor(emp),
    temp = as.factor(temp)
    ) |>
  # Assigning easily readable labels to all variables, using factor() and
   ↪  fct_recode() in a mutate() call.
   mutate(
     sexmig = fct_recode(sexmig,
                         "NonEUMigWo" = "1",
                         "EUMigWo" = "2",
                         "NatWo" = "3",
                         "NonEUMigMe" = "4",
                         "EUMigMe" = "5",
                         "NatMe" = "6"
                         ),

    isco_fct = factor(isco,
                  levels = c("1", "2", "3", "4", "5", "6", "7", "8", "9"),
                  labels = c("Managers", "Professionals", "Technicians",
                             "Clerical workers", "Service workers",
                             "Skilled workers",
                             "Craft and related trades workers",
```

```
                          "Operators", "Elementary workers")),

    PL050_fct = factor(PL050,
                levels = c("11", "12", "13", "21", "22", "23", "24",
                           "31", "32", "33", "34", "41", "42", "51", "52",
                           "61", "71", "72", "73", "74", "81", "82", "83",
                           "91", "92", "93"),
                labels = c("Legislators, senior officials and managers",
                           "Corporate Managers",
                           "Managers of small enterprises",
                           "Physical, mathematical and engineering
                            ↪  professionals",
                           "Life science and health professionals",
                           "Teaching professionals",
                           "Other professionals",
                           "Physical and engineering science associate
                            ↪  professionals",
                           "Life science and health associate
                            ↪  professionals",
                           "Teaching associate professionals",
                           "Other associate professionals",
                           "Office clerks", "Customer services clerks",
                           "Personal and protective services workers",
                           "Models, salespersons and demonstrators",
                           "Skilled agricultural and fishery workers",
                           "Extraction and building trades workers",
                           "Metal, machinery and related trades workers",
                           "Precision, handicraft, craft printing and
                            ↪  related trades workers",
                           "Other craft and related trades workers",
                           "Stationary-plant and related operators",
                           "Machine operators and assemblers",
                           "Drivers and mobile plant operators",
                           "Sales and services elementary occupations",
                           "Agricultural, fishery and related labourers",
                           "Labourers in mining, construction,
                            ↪  manufacturing and transport")),

    PL051_fct = factor(PL051,
                levels = c("11", "12", "13", "14",
                           "21", "22", "23", "24", "25", "26",
                           "31", "32", "33", "34", "35",
```

```
                          "41", "42", "43", "44",
                          "51", "52", "53", "54",
                          "61", "62", "63",
                          "71", "72", "73", "74", "75",
                          "81", "82", "83",
                          "91", "92", "93", "94", "95", "96"),

                 labels = c("Chief executives, senior officials and
                  ↪  legislators",
                          "Administrative and commercial managers",
                          "Production and specialised services managers",
                          "Hospitality, retail and other services
                           ↪  managers",
                          "Science and engineering professionals",
                          "Health professionals",
                          "Teaching professionals",
                          "Business and administration professionals",
                          "Information and communications technology
                           ↪  professionals",
                          "Legal, social and cultural professionals",
                          "Science and engineering associate
                           ↪  professionals",
                          "Health associate professionals",
                          "Business and administration associate
                           ↪  professionals",
                          "Legal, social, cultural and related associate
                           ↪  professionals",
                          "Information and communications technicians",
                          "General and keyboard clerks",
                          "Customer services clerks",
                          "Numerical and material recording clerks",
                          "Other clerical support workers",
                          "Personal service workers",
                          "Sales workers",
                          "Personal care workers",
                          "Protective services workers",
                          "Market-oriented skilled agricultural workers",
                          "Market-oriented skilled forestry, fishery and
                           ↪  hunting workers",
                          "Subsistence farmers, fishers, hunters and
                           ↪  gatherers",
                          "Building and related trades workers, excluding
                           ↪  electricians",
```

```
                                "Metal, machinery and related trades workers",
                                "Handicraft and printing workers",
                                "Electrical and electronic trades workers",
                                "Food processing, wood working, garment and
                                 ↪  other craft and related trades workers",
                                "Stationary plant and machine operators",
                                "Assemblers",
                                "Drivers and mobile plant operators",
                                "Cleaners and helpers",
                                "Agricultural, forestry and fishery labourers",
                                "Labourers in mining, construction,
                                 ↪  manufacturing and transport",
                                "Food preparation assistants",
                                "Street and related sales and service workers",
                                "Refuse workers and other elementary
                                 ↪  workers")),

  occrit = fct_recode(occrit,
                      "Non-essential occupation" = "0",
                      "Essential occupation" = "1"
  ),

  emp = fct_recode(emp,
                   "Employed" = "1",
                   "Unemployed" = "2",
                   "Inactive" = "3"
  ),

  temp = fct_recode(temp,
                    "Permanent job" = "1",
                    "Temporary job" = "2"
  )
)
```

2.3 Generating the analytic samples

We utilise two analytic samples of individuals aged 16–64, excluding the self-employed, retirees, students, and military personnel. These comprise, respectively, the full labour force and only those in paid employment. First, we filter out all self-employed respondents (*PL040*) from `d`.

```
# Filtering out all self-employed respondents ( 3), using the filter()
 ↪  `tidyverse` function.
d <- d |> filter(PL040 %in% c(3, NA)) |> select(-PL040)

# N = 2.140.998 is the provisional sample size of `d`.
# 2.442.653 - 2.140.998 = 301.655 are self-employed or family-working
 ↪  individuals.
# 301.655 / 2.442.653 = 12.35% of total cases are removed.
```

Second, we retain only employed individuals (*emp*) in a separate analytic subsample, referred to as `d_emp`. The variables *temp* and *PL060* are preserved exclusively within this subsample.

```
# Creating a separate subsample containing only employed respondents, using
 ↪  the filter() `tidyverse` function.

d_emp <- d |> filter(emp %in% c("Employed"))

# N = 1.563.301 is the provisional sample size of `d_emp`.
# 2.140.998 - 1.563.301 = 577.697 individuals are unemployed, inactive, or
 ↪  missing employment status.
# 577.697 / 2.140.998 = 26.98% of total cases are removed.

# Dropping temp and PL060 from the main sample, using the select()
 ↪  `tidyverse` function.
d <- d |> select(-PL060, -temp)
```

Third, we exclude all individuals reporting more than 70 working hours per week (*PL060*) from the `d_emp` subsample and apply casewise deletion to both `d` and `d_emp`.

```
# Applying casewise deletion with the filter() `tidyverse` function, using
 ↪  is.na() to identify missing values.

# Notes:
# PL050 is available only in 2007-2010, PL051 only in 2011-2022.
# Filtering conditions must account for this distinction.
# drop_na() removes remaining missing values, excluding PL050/PL051 and their
 ↪  recoded versions.
# relocate() is used to reorder factor variables next to their originals for
 ↪  added clarity.

# Main analytic sample (`d`).
```

```r
d <- d |> filter(
  !(year %in% c("2007", "2008", "2009", "2010") & is.na(PL050)),
  !(year %in% c("2011", "2012", "2015", "2018", "2019", "2020",
                "2021", "2022") & is.na(PL051))) |>
  drop_na(all_of(setdiff(names(d), c("PL050", "PL051", "PL050_fct",
   ↪  "PL051_fct")))) |>
  relocate(isco_fct, .after = isco) |>
  relocate(PL050_fct, .after = PL050) |>
  relocate(PL051_fct, .after = PL051)

# N = 1.838.684 is the final sample size of `d`.
# (2.140.998 - 1.838.684) / 2.140.998 = 14.12% of incomplete cases are
 ↪  removed.

# Employed subsample (`d_emp`).
# Adding a filter() condition based on PL060, to exclude outliers for hours
 ↪  worked per week.
d_emp <- d_emp |> filter(
  !(year %in% c("2007", "2008", "2009", "2010") & is.na(PL050)),
  !(year %in% c("2011", "2012", "2015", "2018", "2019", "2020",
                "2021", "2022") & is.na(PL051))) |>
  drop_na(all_of(setdiff(names(d_emp), c("PL050", "PL051", "PL050_fct",
   ↪  "PL051_fct")))) |>
  filter(PL060 <= 70) |>
  relocate(isco_fct, .after = isco) |>
  relocate(PL050_fct, .after = PL050) |>
  relocate(PL051_fct, .after = PL051)

# N = 1.296.594 is the final sample size of `d_emp`.
# (1.563.301 - 1.296.594) / 1.563.301 = 17.06% of incomplete cases or
 ↪  outliers are removed.
```

Finally, the tidied samples `d` and `d_emp` are saved using the `write_dta` function from the `haven` package.

```r
# Saving the tidied samples as Stata data files (.dta).
write_dta(d, "Data/d.dta")
write_dta(d_emp, "Data/d_emp.dta")
```

3. Data Visualisation

Now, we turn to data visualisation. Our book chapter includes four main figures:

2.1. *Yearly Trends in Rates of Employment, Unemployment, and Inactivity by Sex and Migrant Background.*

2.2. *Yearly Trends in Average Weekly Working Hours by Sex and Migrant Background.*

2.3. *Yearly Trends in Temporary Employment Rates by Sex and Migrant Background.*

2.4. *Yearly Trends in Employment in Essential Occupations by Sex and Migrant Background.*

It is important to note that **Figure 2.1** is submitted as three separate panels, that is, **2.1a** (*Employment*), **2.1b** (*Unemployment*), and **2.1c** (*Inactivity*) on account of editorial constraints.

Our book chapter also contains two main tables:

2.1. *Changes in Occupational Concentration (1-digit ISCO) during the Great Recession, by Sex and Migrant Background.*

2.2. *Changes in Occupational Concentration (1-digit ISCO) during the COVID-19 Pandemic, by Sex and Migrant Background.*

The **Online Appendix** is comprised by ten supplementary tables:

A2.1. *Changes in Rates of Employment, Unemployment, and Inactivity, by Sex and Migrant Background.*

A2.2. *Yearly Trends in Rates of Employment, Unemployment, and Inactivity, by Sex and Migrant Background.*

A2.3. *Changes in Weekly Working Hours and Rates of Temporary Employment, by Sex and Migrant Background.*

A2.4. *Yearly Trends in Average Weekly Working Hours by Sex and Migrant Background.*

A2.5. *Yearly Trends in Rates of Temporary and Permanent Employment, by Sex and Migrant Background.*

A2.6. *Yearly Trends in Mutual Information Index of Occupational Segregation, by Sex and Migrant Background.*

A2.7a. *Yearly Trends in Occupational Concentration (1-digit ISCO) among Women, by Migrant Background.*

A2.7b. *Yearly Trends in Occupational Concentration (1-digit ISCO) among Men, by Migrant Background.*

A2.8. *Yearly Trends in Rates of Employment in Essential Occupations, by Sex and Migrant Background.*

A2.9. *Changes in Rates of Employment in Essential Occupations, by Sex and Migrant Background.*

3.1 Table A2.1 / Table A2.2 / Figure 2.1a / Figure 2.1b / Figure 2.1c

Table A2.1, as well as **Figures 2.1a** (*Employed*), **2.1b** (*Unemployed*), and **2.1c** (*Inactive*), all derive from the raw format tibble that synthesises the information contained in **Table A2.2**.

```
tablea22 <- d |>
  # Only the relevant variables and the design weights are included in the
   ↪ raw format tibble.
  select(year, sexmig, emp, RB050) |>
  # Recoding all sex and migrant background categories to enhance their
   ↪ readability, using fct_recode() in a mutate() call.
  mutate(
    sexmig = fct_recode(sexmig,
                        "Non-EU Migrant Women" = "NonEUMigWo",
                        "EU Migrant Women" = "EUMigWo",
                        "Native Women" = "NatWo",
                        "Non-EU Migrant Men" = "NonEUMigMe",
                        "EU Migrant Men" = "EUMigMe",
                        "Native Men" = "NatMe")) |>
  # Applying design weights with the as_survey() function from the `srvyr`
   ↪ package.
  as_survey(weights = RB050) |>
  # Grouping by year, sex and migrant background, and employment status.
  group_by(year, sexmig, emp) |>
  # Counting the weighted number of respondents within each group.
  summarise(weighted_n = survey_total()) |>
  # Transforming the weighted counts into percentages and rounding at two
   ↪ decimal places.
  mutate(perc = round(weighted_n / sum(weighted_n) * 100, 2)) |>
  # Resetting the grouping.
  ungroup() |>
  # Removing the weighted counts and their associated standard errors.
  select(-weighted_n, -weighted_n_se)
```

Table A2.2 is appropriately styled and saved as a `.png` file using the `gt` package.

```r
# Table A2.2. Yearly Trends in Rates of Employment, Unemployment, and
 ↪  Inactivity, by Sex and Migrant Background.

tablea22_gt <- tablea22 |>
  # Recoding all percentages and years as strings in a mutate() call, so that
   ↪  `gt` does not mess with their visualisation.
  mutate(perc = sprintf("%.2f%%", perc),
         year = as.character(year)) |>
  # Pivoting the table to a wider format, assigning a separate column of
   ↪  percentages to each year.
  pivot_wider(names_from = year, values_from = perc) |>
  # Building the `gt` table, setting row names as employment status
   ↪  categories, and grouping rows by sex and migrant background.
  gt(rowname_col = "emp", groupname_col = "sexmig") |>
  # Assigning an appropriate title and subtitle to the `gt` table.
  tab_header(
    title = md("**Table A2.2. Yearly Trends in Rates of Employment,
     ↪  Unemployment, and Inactivity**"),
    subtitle = md("*by Sex and Migrant Background*")
  ) |>
  # Setting column labels as strings, so that `gt` does not mess with their
   ↪  visualisation.
  cols_label(
    `2007` = "2007", `2008` = "2008", `2009` = "2009",
    `2010` = "2010", `2011` = "2011", `2012` = "2012",
    `2015` = "2015", `2018` = "2018", `2019` = "2019",
    `2020` = "2020", `2021` = "2021", `2022` = "2022"
  ) |>
  # Imposing the FiveThirtyEight (538) theme on the `gt` table.
  gt_theme_538() |>
  # Fixing the stubhead label to make it clear that each column represents a
   ↪  different year.
  tab_stubhead(label = "Year") |>
  # Styling the stubhead cells to make them prettier.
  tab_style(
    style = list(
      cell_text(
        weight = "bold",
        style = "italic",
        align = "center",
        v_align = "middle"
      ),
```

```
    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_stubhead()
  ) |>
# Styling the title cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_title()
) |>
# Styling the column labels cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle"
    ),

    cell_fill(
    color = "#D6D6D6"
    )
  ),
  locations = cells_column_labels()
) |>
# Styling the row grouping cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      style = "italic",
      align = "center"
    ),
```

```
      cell_fill(
        color = "#D6D6D6"
      )
    ),
    locations = cells_row_groups()
  ) |>
  # Styling the body cells to make them prettier.
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_body()
  ) |>
  # Setting other general table options for prettier visualisation.
  tab_options(
    table.font.size = px(14),
    heading.align = "center",
    row_group.font.weight = "bold",
    heading.subtitle.font.size = px(16),
    heading.title.font.size = px(18),
    column_labels.hidden = FALSE
  ) |>
  # Setting the source note to comply with editorial requirements.
  tab_source_note(
    source_note = "Source: Own calculations, EU-SILC 2007-22 (N = 1.838.684);
    ↪  weighted sample."
  ) |>
  # Styling the source notes cells to make them prettier.
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_source_notes()
  )

# Saving the `gt` table as a `.png` file.
gtsave(tablea22_gt, "Visuals/Tables/TableA22.png")
```

Table A2.2. Yearly Trends in Rates of Employment, Unemployment, and Inactivity

by Sex and Migrant Background

Year	*2007*	*2008*	*2009*	*2010*	*2011*	*2012*	*2015*	*2018*	*2019*	*2020*	*2021*	*2022*
Non-EU Migrant Women												
Employed	69.07%	70.03%	63.08%	62.19%	57.71%	58.10%	56.57%	65.32%	64.88%	60.98%	72.82%	77.02%
Unemployed	11.53%	9.62%	16.14%	16.46%	20.85%	22.12%	20.57%	16.00%	15.31%	20.76%	15.94%	11.94%
Inactive	19.40%	20.34%	20.78%	21.34%	21.44%	19.78%	22.87%	18.67%	19.81%	18.26%	11.24%	11.04%
EU Migrant Women												
Employed	73.45%	71.50%	68.09%	66.88%	67.88%	67.47%	70.66%	77.14%	76.31%	72.67%	78.54%	82.76%
Unemployed	6.63%	7.89%	13.91%	11.18%	13.61%	15.77%	15.74%	10.50%	9.39%	13.92%	11.65%	8.46%
Inactive	19.92%	20.61%	18.00%	21.93%	18.50%	16.76%	13.60%	12.36%	14.31%	13.41%	9.81%	8.78%
Native Women												
Employed	74.95%	75.76%	74.41%	73.42%	73.53%	73.44%	74.26%	77.84%	78.62%	78.49%	83.54%	85.48%
Unemployed	7.94%	6.83%	8.92%	9.82%	9.81%	11.43%	11.29%	9.35%	8.83%	9.25%	8.38%	7.01%
Inactive	17.11%	17.41%	16.67%	16.75%	16.65%	15.14%	14.45%	12.81%	12.56%	12.27%	8.08%	7.51%
Non-EU Migrant Men												
Employed	82.40%	79.12%	67.82%	68.00%	66.09%	64.13%	71.09%	75.85%	75.56%	71.63%	83.40%	83.66%
Unemployed	14.37%	16.56%	27.80%	27.38%	28.42%	30.39%	22.36%	19.84%	18.70%	21.47%	13.85%	13.76%
Inactive	3.22%	4.31%	4.38%	4.62%	5.49%	5.48%	6.56%	4.31%	5.75%	6.90%	2.75%	2.58%
EU Migrant Men												
Employed	87.07%	87.52%	81.07%	77.19%	77.60%	76.87%	80.63%	82.04%	84.55%	82.74%	88.58%	90.73%
Unemployed	7.77%	7.93%	15.73%	17.49%	18.02%	19.38%	15.36%	13.11%	11.14%	11.97%	10.00%	7.58%
Inactive	5.16%	4.56%	3.20%	5.32%	4.38%	3.75%	4.02%	4.85%	4.32%	5.29%	1.42%	1.69%
Native Men												
Employed	87.64%	87.59%	83.94%	82.66%	82.79%	81.80%	82.60%	86.06%	86.32%	85.39%	90.53%	92.01%
Unemployed	7.06%	7.03%	10.62%	11.58%	11.43%	12.49%	12.09%	8.97%	8.40%	8.94%	8.28%	6.80%
Inactive	5.30%	5.37%	5.45%	5.76%	5.78%	5.71%	5.31%	4.98%	5.27%	5.67%	1.19%	1.19%

Source: Own calculations, EU-SILC 2007-22 (N = 1.838.684); weighted sample.

Figure 2.1a is appropriately styled and saved as a `.svg` file using the `ggplot2` package.

```
# Figure 2.1a. Yearly Trends in Rates of Employment, by Sex and Migrant
 ↪  Background.

figure21a <- tablea22 |>
  # Recoding all years as numeric in a mutate() call, so that `ggplot2` can
   ↪  generate a line plot.
  mutate(year = as.numeric(year)) |>
  # Keeping employed respondents only, using the filter() `tidyverse`
   ↪  function.
  filter(emp == "Employed") |>
  # Generating a lineplot of rates of employment by year, sex, and migrant
   ↪  background.
  ggplot(aes(x = year, y = perc, linetype = sexmig, color = sexmig)) +
  # Setting an appropriate line width to facilitate the figure's
   ↪  interpretability.
  geom_line(linewidth = 1) +
  # Manually annotating rectangles of differing grey shades to highlight the
   ↪  Great Recession and the COVID-19 pandemic.
  annotate("rect", xmin = 2, xmax = 3, ymin = -Inf, ymax = Inf,
           alpha = 0.3, fill = "#A3A3A3") +
  annotate("rect", xmin = 3, xmax = 6, ymin = -Inf, ymax = Inf,
           alpha = 0.3, fill = "#C2C2C2") +
  annotate("rect", xmin = 10, xmax = 11, ymin = -Inf, ymax = Inf,
           alpha = 0.3, fill = "#A3A3A3") +
  annotate("rect", xmin = 11, xmax = 12, ymin = -Inf, ymax = Inf,
           alpha = 0.3, fill = "#C2C2C2") +
  # Assigning appropriate labels to all rectangles to highlight the Great
   ↪  Recession and the COVID-19 pandemic.
  geom_richtext(data = tibble(
    year = c(2.5, 4.5, 10.5, 11.5),
    perc = c(94, 94, 94, 94),
    label = c("Great Recession<br>(short-term)", "Great
     ↪  Recession<br>(long-term)",
              "COVID-19<br>(short-term)", "COVID-19<br>(long-term)"),
    emp = "Employed"
    ),
    aes(x = year, y = perc, label = label),
    inherit.aes = FALSE, size = 3.5, color = "black", fontface = "bold") +
  # Setting a continuous year-by-year scale on the x-axis.
  scale_x_continuous(breaks = c(1:12),
                     labels = c("2007", "2008", "2009", "2010", "2011",
                      ↪  "2012",
```

```
                                "2015", "2018", "2019", "2020", "2021",
                                ↪  "2022")
                    ) +
  # Setting a continuous percentage scale on the y-axis.
  scale_y_continuous(breaks = seq(0, 95, 1)) +
  # Setting combinations of grey shades and line types to highlight sex and
   ↪  migrant background categories.
  scale_linetype_manual(values = c("solid", "dotted", "twodash",
                                   "solid", "dotted", "twodash")) +
  scale_color_manual(values = c("#000000", "#000000", "#000000",
                                "#999999", "#999999", "#999999")) +
  # Styling axes and legends to make the figure look prettier.
  guides(color = guide_legend(byrow = TRUE), linetype = guide_legend(byrow =
   ↪  TRUE)) +
  labs(
    title = "",
    subtitle = "",
    x = "",
    y = "% of Respondents",
    color = "",
    linetype = ""
  ) +
  # Styling the overall theme to make the figure look prettier.
  theme_bw(base_size = 14) +
  theme(
    plot.title.position = "plot",
    plot.title = element_text(hjust = 0.5, face = "bold", size = 18),
    plot.subtitle = element_text(hjust = 0.5, face = "italic", size = 14),
    legend.position = "bottom",
    strip.text = element_text(face = "bold")
  ) +
  # Specifying a facet_grid() to wrap the figure inside an "Employed" panel.
  facet_grid(rows = vars(emp), scales = "free")

# Saving the `ggplot2` figure as a `.svg` file.
ggsave(filename = "Visuals/Figures/Figure21a.svg",
       plot = figure21a, width = 12, height = 10, dpi = 900)
```

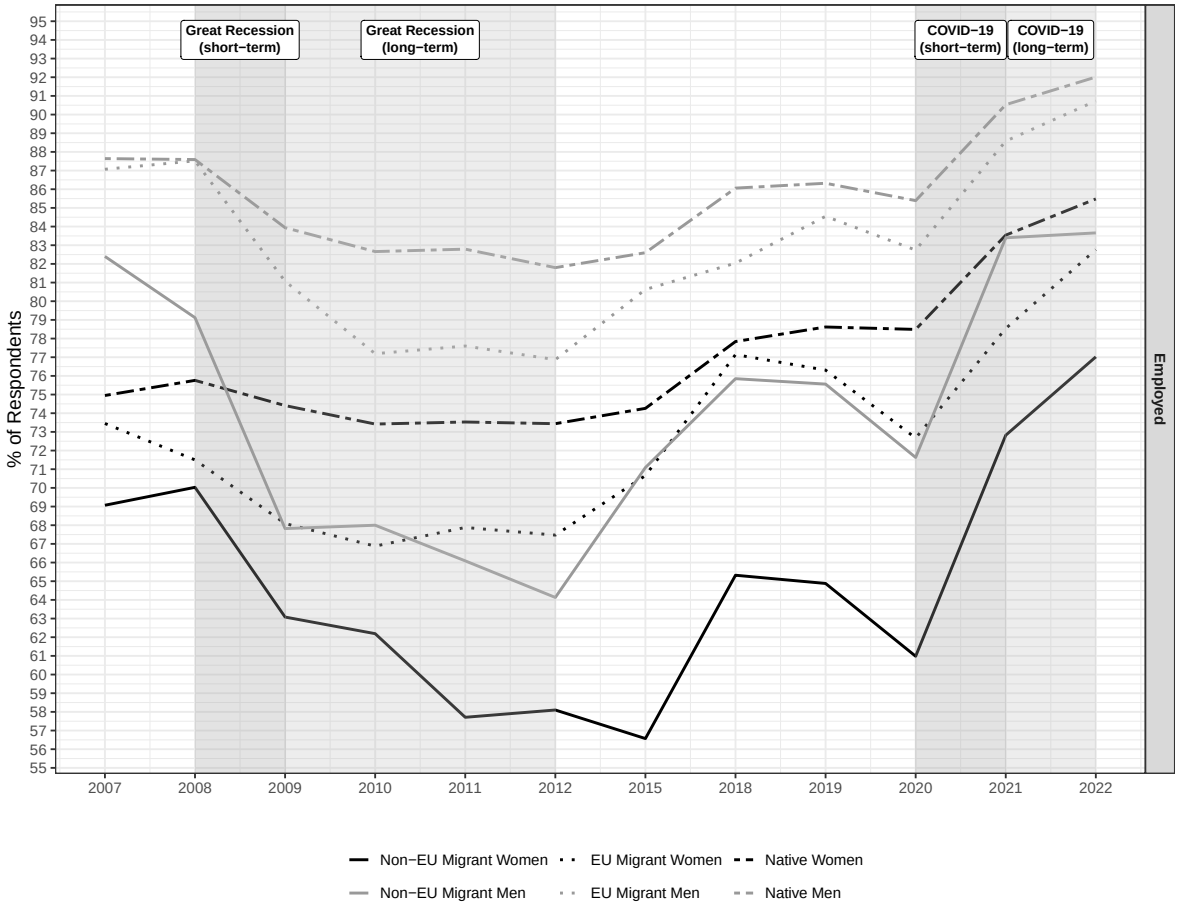

Figure 2.1a. *Yearly Trends in Rates of Employment, by Sex and Migrant Background.*

Figure 2.1b is appropriately styled and saved as a `.svg` file using the `ggplot2` package.

```
# Figure 2.1b. Yearly Trends in Rates of Unemployment, by Sex and Migrant
 ↪  Background.

figure21b <- tablea22 |>
  # Recoding all years as numeric in a mutate() call, so that `ggplot2` can
   ↪  generate a line plot.
  mutate(year = as.numeric(year)) |>
  # Keeping unemployed respondents only using the filter() `tidyverse`
   ↪  function.
  filter(emp == "Unemployed") |>
```

```
# Generating a lineplot of rates of unemployment by year, sex, and migrant
 ↪  background.
ggplot(aes(x = year, y = perc, linetype = sexmig, color = sexmig)) +
# Setting an appropriate line width to facilitate the figure's
 ↪  interpretability.
geom_line(linewidth = 1) +
# Manually annotating rectangles of differing grey shades to highlight the
 ↪  Great Recession and the COVID-19 pandemic.
annotate("rect", xmin = 2, xmax = 3, ymin = -Inf, ymax = Inf,
         alpha = 0.3, fill = "#A3A3A3") +
annotate("rect", xmin = 3, xmax = 6, ymin = -Inf, ymax = Inf,
         alpha = 0.3, fill = "#C2C2C2") +
annotate("rect", xmin = 10, xmax = 11, ymin = -Inf, ymax = Inf,
         alpha = 0.3, fill = "#A3A3A3") +
annotate("rect", xmin = 11, xmax = 12, ymin = -Inf, ymax = Inf,
         alpha = 0.3, fill = "#C2C2C2") +
# Assigning appropriate labels to all rectangles to highlight the Great
 ↪  Recession and the COVID-19 pandemic.
geom_richtext(data = tibble(
  year = c(2.5, 4.5, 10.5, 11.5),
  perc = c(31, 31, 31, 31),
  label = c("Great Recession<br>(short-term)", "Great
   ↪  Recession<br>(long-term)",
            "COVID-19<br>(short-term)", "COVID-19<br>(long-term)"),
  emp = "Unemployed"
  ),
  aes(x = year, y = perc, label = label),
  inherit.aes = FALSE, size = 3.5, color = "black", fontface = "bold") +
# Setting a continuous year-by-year scale on the x-axis.
scale_x_continuous(breaks = c(1:12),
                   labels = c("2007", "2008", "2009", "2010", "2011",
                    ↪  "2012",
                              "2015", "2018", "2019", "2020", "2021",
                               ↪  "2022")
                   ) +
# Setting a continuous percentage scale on the y-axis.
scale_y_continuous(breaks = seq(0, 95, 1)) +
# Setting combinations of grey shades and line types to highlight sex and
 ↪  migrant background categories.
scale_linetype_manual(values = c("solid", "dotted", "twodash",
                                 "solid", "dotted", "twodash")) +
scale_color_manual(values = c("#000000", "#000000", "#000000",
```

```
                                  "#999999", "#999999", "#999999")) +
  # Styling axes and legends to make the figure look prettier.
  guides(color = guide_legend(byrow = TRUE), linetype = guide_legend(byrow =
   ↪  TRUE)) +
  labs(
    title = "",
    subtitle = "",
    x = "",
    y = "% of Respondents",
    color = "",
    linetype = ""
  ) +
  # Styling the overall theme to make the figure look prettier.
  theme_bw(base_size = 14) +
  theme(
    plot.title.position = "plot",
    plot.title = element_text(hjust = 0.5, face = "bold", size = 18),
    plot.subtitle = element_text(hjust = 0.5, face = "italic", size = 14),
    legend.position = "bottom",
    strip.text = element_text(face = "bold")
  ) +
   # Specifying a facet_grid() to wrap the figure inside an "Unemployed"
    ↪  panel.
  facet_grid(rows = vars(emp), scales = "free")

# Saving the `ggplot2` figure as a `.svg` file.
ggsave(filename = "Visuals/Figures/Figure21b.svg",
       plot = figure21b, width = 12, height = 10, dpi = 900)
```

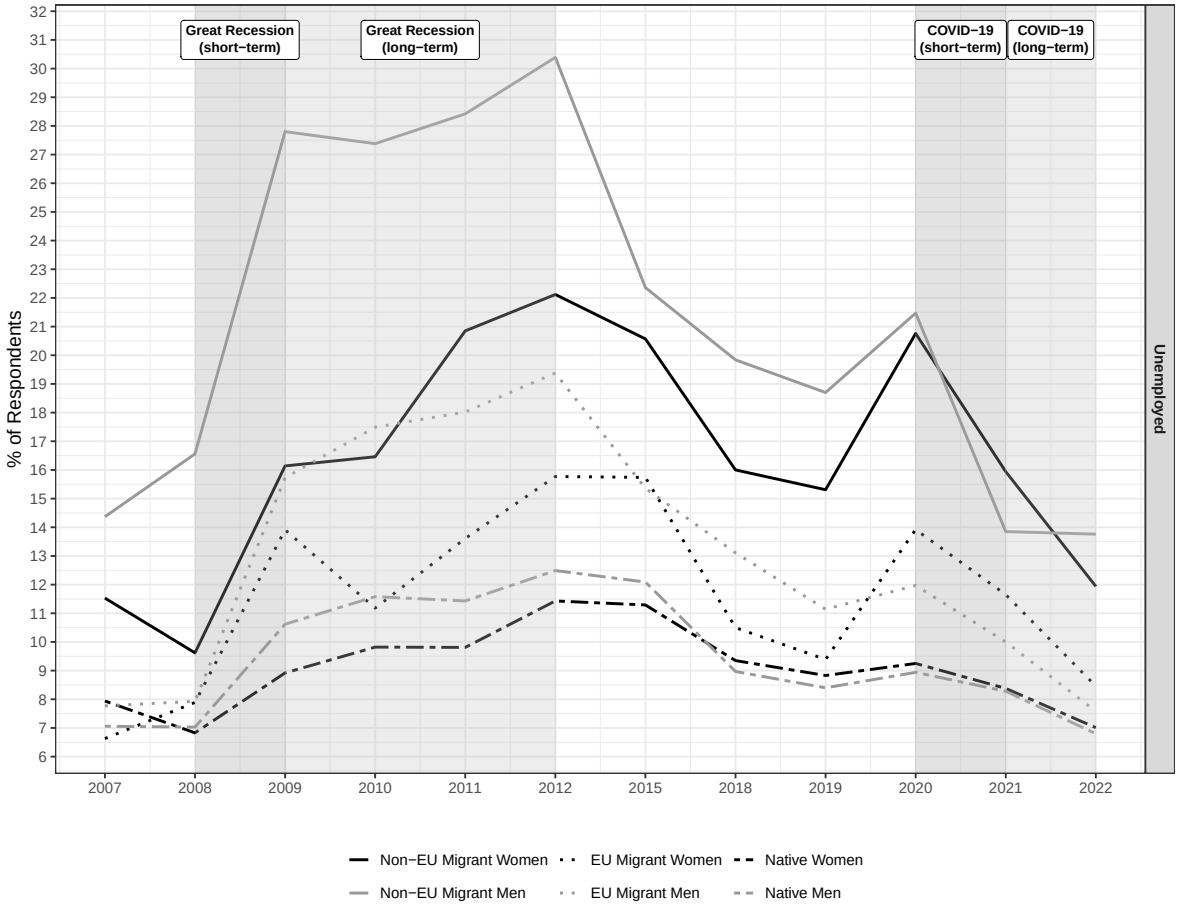

Figure 2.1b. *Yearly Trends in Rates of Unemployment, by Sex and Migrant Background.*

Figure 2.1c is appropriately styled and saved as a `.svg` file using the `ggplot2` package.

```
# Figure 2.1c. Yearly Trends in Rates of Inactivity, by Sex and Migrant
 ↪  Background.

figure21c <- tablea22 |>
  # Recoding all years as numeric in a mutate() call, so that `ggplot2` can
   ↪  generate a line plot.
  mutate(year = as.numeric(year)) |>
  # Keeping unemployed respondents only using the filter() `tidyverse`
   ↪  function.
  filter(emp == "Inactive") |>
```

```
# Generating a lineplot of rates of inactivity by year, sex, and migrant
 ↪ background.
ggplot(aes(x = year, y = perc, linetype = sexmig, color = sexmig)) +
# Setting an appropriate line width to facilitate the figure's
 ↪ interpretability.
geom_line(linewidth = 1) +
# Manually annotating rectangles of differing grey shades to highlight the
 ↪ Great Recession and the COVID-19 pandemic.
annotate("rect", xmin = 2, xmax = 3, ymin = -Inf, ymax = Inf,
         alpha = 0.3, fill = "#A3A3A3") +
annotate("rect", xmin = 3, xmax = 6, ymin = -Inf, ymax = Inf,
         alpha = 0.3, fill = "#C2C2C2") +
annotate("rect", xmin = 10, xmax = 11, ymin = -Inf, ymax = Inf,
         alpha = 0.3, fill = "#A3A3A3") +
annotate("rect", xmin = 11, xmax = 12, ymin = -Inf, ymax = Inf,
         alpha = 0.3, fill = "#C2C2C2") +
# Assigning appropriate labels to all rectangles to highlight the Great
 ↪ Recession and the COVID-19 pandemic.
geom_richtext(data = tibble(
  year = c(2.5, 4.5, 10.5, 11.5),
  perc = c(24, 24, 24, 24),
  label = c("Great Recession<br>(short-term)", "Great
   ↪ Recession<br>(long-term)",
            "COVID-19<br>(short-term)", "COVID-19<br>(long-term)"),
  emp = "Inactive"
  ),
  aes(x = year, y = perc, label = label),
  inherit.aes = FALSE, size = 3.5, color = "black", fontface = "bold") +
# Setting a continuous year-by-year scale on the x-axis.
scale_x_continuous(breaks = c(1:12),
                   labels = c("2007", "2008", "2009", "2010", "2011",
                    ↪ "2012",
                              "2015", "2018", "2019", "2020", "2021",
                               ↪ "2022")
                   ) +
# Setting a continuous percentage scale on the y-axis.
scale_y_continuous(breaks = seq(0, 95, 1)) +
# Setting combinations of grey shades and line types to highlight sex and
 ↪ migrant background categories.
scale_linetype_manual(values = c("solid", "dotted", "twodash",
                                 "solid", "dotted", "twodash")) +
scale_color_manual(values = c("#000000", "#000000", "#000000",
```

```
                                  "#999999", "#999999", "#999999")) +
  # Styling axes and legends to make the figure look prettier.
  guides(color = guide_legend(byrow = TRUE), linetype = guide_legend(byrow =
   ↪  TRUE)) +
  labs(
    title = "",
    subtitle = "",
    x = "",
    y = "% of Respondents",
    color = "",
    linetype = ""
  ) +
  # Styling the overall theme to make the figure look prettier.
  theme_bw(base_size = 14) +
  theme(
    plot.title.position = "plot",
    plot.title = element_text(hjust = 0.5, face = "bold", size = 18),
    plot.subtitle = element_text(hjust = 0.5, face = "italic", size = 14),
    legend.position = "bottom",
    strip.text = element_text(face = "bold")
  ) +
  # Specifying a facet_grid() to wrap the figure inside an "Inactive" panel.
  facet_grid(rows = vars(emp), scales = "free")

# Saving the `ggplot2` figure as a `.svg` file.
ggsave(filename = "Visuals/Figures/Figure21c.svg",
       plot = figure21c, width = 12, height = 10, dpi = 900)
```

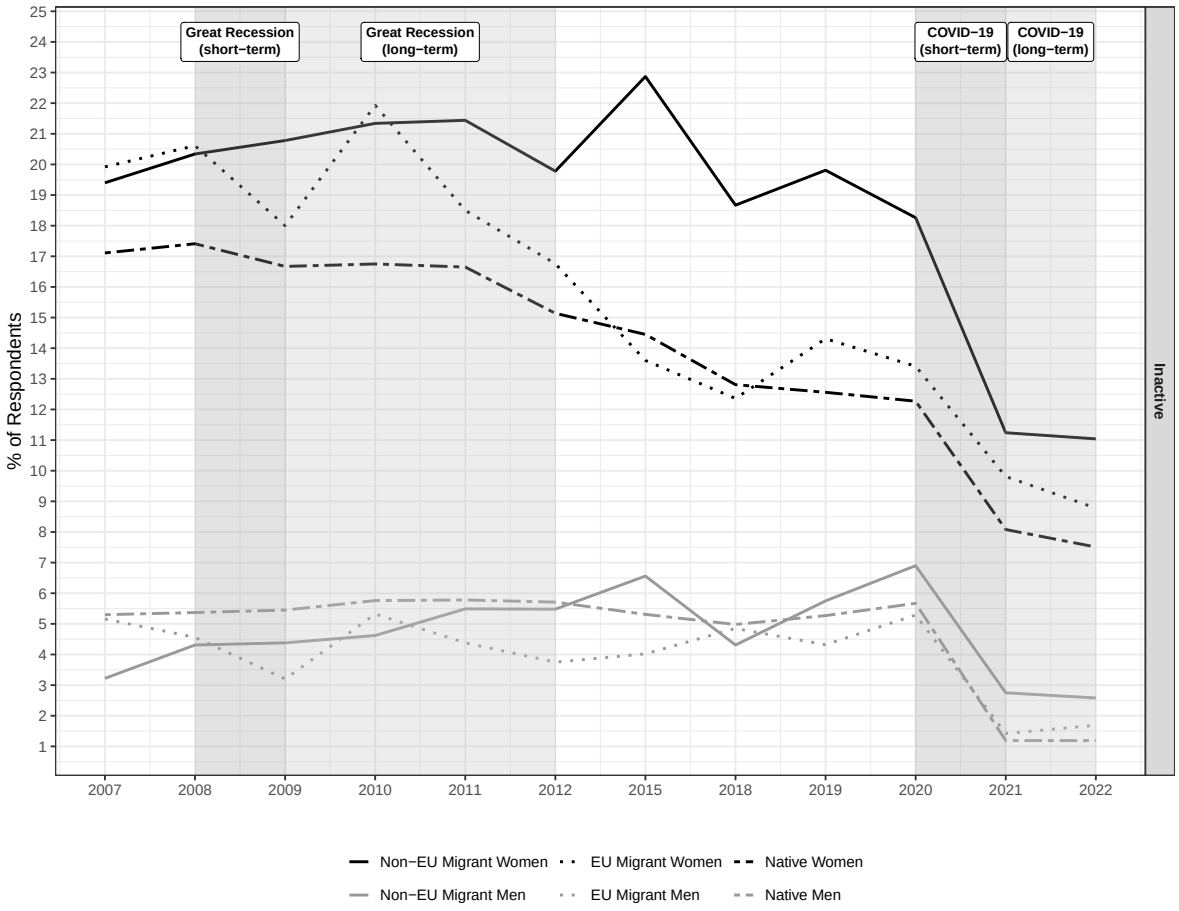

Figure 2.1c. *Yearly Trends in Rates of Inactivity, by Sex and Migrant Background.*

Table A2.1 is appropriately styled and saved as a `.png` file using the `gt` package.

```
# Table A2.1. Changes in Rates of Employment, Unemployment, and Inactivity,
 ↪  by Sex and Migrant Background.

tablea21_gt <- tablea22 |>
  # Isolating all years delimiting the Great Recession and the COVID-19
   ↪  pandemic, using the filter() `tidyverse` function.
  filter(year %in% c("2007", "2009", "2012", "2019", "2020", "2022")) |>
  # Pivoting the table to a wider format, assigning a separate column of
   ↪  percentages to each year.
  pivot_wider(names_from = year, values_from = perc) |>
```

```
# Calculating all the short-term and long-term percentage changes.
mutate(
  change_07_09 = (`2009` - `2007`), # Great Recession (short-term).
  change_07_12 = (`2012` - `2007`), # Great Recession (long-term).
  change_19_20 = (`2020` - `2019`), # COVID-19 pandemic (short-term).
  change_19_22 = (`2022` - `2019`) # COVID-19 pandemic (long-term).
) |>
# Pivoting the table to a wider format, assigning a single column to all
 ↪  percentage changes.
pivot_longer(cols = c(change_07_09, change_07_12,
                      change_19_20, change_19_22),
             names_to = "period",
             values_to = "change") |>
# Dropping all superfluous variables, using the select() `tidyverse`
 ↪  function.
select(sexmig, emp, period, change) |>
# Recoding all variable labels to enhance their readability, using
 ↪  fct_recode() in a mutate() call.
mutate(
  period = fct_recode(as.factor(period),
                      "Great Recession (2007-09, short-term)" =
                        "change_07_09",
                      "Great Recession (2007-12, long-term)" =
                        "change_07_12",
                      "COVID-19 (2019-20, short-term)" =
                        "change_19_20",
                      "COVID-19 (2019-22, long-term)" =
                        "change_19_22")) |>
# Recoding all percentage changes as strings, so that `gt` does not mess
 ↪  with their visualisation.
mutate(change = if_else(change > 0,
                        sprintf("+%.2f%%", change),
                        sprintf("%.2f%%", change))) |>
# Pivoting the table to a wider format, assigning a separate column of
 ↪  percentage changes to each employment status category.
pivot_wider(names_from = emp, values_from = change) |>
# Building the `gt` table, setting row names as time periods, and grouping
 ↪  rows by sex and migrant background.
gt(rowname_col = "period", groupname_col = "sexmig") |>
# Assigning an appropriate title and subtitle to the `gt` table.
tab_header(
  title = md("**Table A2.1. Changes in Employment, Unemployment, and
   ↪  Inactivity**"),
```

```
  subtitle = md("*by Sex and Migrant Background*")
) |>
# Imposing the FiveThirtyEight (538) theme on the `gt` table.
gt_theme_538() |>
# Fixing the stubhead label to make it clear that each column represents a
 ↪ different employment status category.
tab_stubhead(label = "Employment Status") |>
# Styling the stubhead cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_stubhead()
  ) |>
# Styling the title cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_title()
) |>
# Styling the column labels cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
```

```
      v_align = "middle",
      whitespace = "pre-wrap",
      transform = "capitalize"
    ),
    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_column_labels()
) |>
# Styling the row grouping cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      style = "italic",
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_row_groups()
) |>
# Styling the body cells to make them prettier.
tab_style(
  style = cell_text(align = "center"),
  locations = cells_body()
) |>
# Setting other general table options for prettier visualisation.
tab_options(
  table.font.size = px(14),
  heading.align = "center",
  row_group.font.weight = "bold",
  heading.subtitle.font.size = px(16),
  heading.title.font.size = px(18)
) |>
# Setting the source note to comply with editorial requirements.
tab_source_note(
  source_note = "Note: Changes are calculated by subtracting the % in
  ↪  2009/12/20/22 from the % in 2007/19.\nSource: Own calculations,
  ↪  EU-SILC 2007-22 (N = 1.838.684); weighted sample."
) |>
```

```
  # Styling the source notes cells to make them prettier.
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_source_notes()
  )

# Saving the `gt` table as a `.png` file.
gtsave(tablea21_gt, "Visuals/Tables/TableA21.png")
```

Table A2.1. Changes in Employment, Unemployment, and Inactivity

by Sex and Migrant Background

Employment Status	*Employed*	*Unemployed*	*Inactive*
Non-EU Migrant Women			
Great Recession (2007-09, short-term)	-5.99%	+4.61%	+1.38%
Great Recession (2007-12, long-term)	-10.97%	+10.59%	+0.38%
COVID-19 (2019-20, short-term)	-3.90%	+5.45%	-1.55%
COVID-19 (2019-22, long-term)	+12.14%	-3.37%	-8.77%
EU Migrant Women			
Great Recession (2007-09, short-term)	-5.36%	+7.28%	-1.92%
Great Recession (2007-12, long-term)	-5.98%	+9.14%	-3.16%
COVID-19 (2019-20, short-term)	-3.64%	+4.53%	-0.90%
COVID-19 (2019-22, long-term)	+6.45%	-0.93%	-5.53%
Native Women			
Great Recession (2007-09, short-term)	-0.54%	+0.98%	-0.44%
Great Recession (2007-12, long-term)	-1.51%	+3.49%	-1.97%
COVID-19 (2019-20, short-term)	-0.13%	+0.42%	-0.29%
COVID-19 (2019-22, long-term)	+6.86%	-1.82%	-5.05%
Non-EU Migrant Men			
Great Recession (2007-09, short-term)	-14.58%	+13.43%	+1.16%
Great Recession (2007-12, long-term)	-18.27%	+16.02%	+2.26%
COVID-19 (2019-20, short-term)	-3.93%	+2.77%	+1.15%
COVID-19 (2019-22, long-term)	+8.10%	-4.94%	-3.17%
EU Migrant Men			
Great Recession (2007-09, short-term)	-6.00%	+7.96%	-1.96%
Great Recession (2007-12, long-term)	-10.20%	+11.61%	-1.41%
COVID-19 (2019-20, short-term)	-1.81%	+0.83%	+0.97%
COVID-19 (2019-22, long-term)	+6.18%	-3.56%	-2.63%
Native Men			
Great Recession (2007-09, short-term)	-3.70%	+3.56%	+0.15%
Great Recession (2007-12, long-term)	-5.84%	+5.43%	+0.41%
COVID-19 (2019-20, short-term)	-0.93%	+0.54%	+0.40%
COVID-19 (2019-22, long-term)	+5.69%	-1.60%	-4.08%

Note: Changes are calculated by subtracting the % in 2009/12/20/22 from the % in 2007/19. Source: Own calculations, EU-SILC 2007-22 (N = 1.838.684); weighted sample.

3.2 Table A2.4 / Figure 2.2

Figure 2.2 derives from the raw format tibble that synthesises the information contained in **Table A2.4**.

```
tablea24 <- d_emp |>
  # Recoding all sex and migrant background categories to enhance their
   ↪ readability, using fct_recode() in a mutate() call.
  mutate(
    sexmig = fct_recode(sexmig,
                        "Non-EU Migrant Women" = "NonEUMigWo",
                        "EU Migrant Women" = "EUMigWo",
                        "Native Women" = "NatWo",
                        "Non-EU Migrant Men" = "NonEUMigMe",
                        "EU Migrant Men" = "EUMigMe",
                        "Native Men" = "NatMe")) |>
  # Applying design weights with the as_survey() function from the `srvyr`
   ↪ package.
  as_survey(weights = RB050) |>
  # Grouping by year and sex and migrant background.
  group_by(year, sexmig) |>
  # Computing the weighted means of weekly working hours within each group.
  summarise(
    PL060 = survey_mean(PL060),
  ) |>
  # Resetting the grouping.
  ungroup() |>
  # Removing the weighted standard errors associated to the means.
  select(-PL060_se)
```

Table A2.4 is appropriately styled and saved as a `.png` file using the `gt` package.

```
# Table A2.4. Yearly Trends in Average Weekly Working Hours by Sex and
 ↪ Migrant Background.

tablea24_gt <- tablea24 |>
  # Recoding all average weekly working hours and years as strings in a
   ↪ mutate() call, so that `gt` does not mess with their visualisation.
  mutate(PL060 = sprintf("%.2fh", PL060),
         year = as.character(year)) |>
  # Pivoting the table to a wider format, assigning a separate column of
   ↪ average weekly working hours to each year.
```

```
pivot_wider(names_from = year, values_from = PL060) |>
# Building the `gt` table, grouping rows by sex and migrant background.
# Row names are left empty, since only employed people are included in the
 ↪  `d_emp` analytic subsamples.
gt(rowname_col = "emp", groupname_col = "sexmig") |>
# Assigning an appropriate title and subtitle to the `gt` table.
tab_header(
  title = md("**Table A2.4. Yearly Trends in Average Weekly Working
   ↪  Hours**"),
  subtitle = md("*by Sex and Migrant Background*")
) |>
# Setting column labels as strings, so that `gt` does not mess with their
 ↪  visualisation.
cols_label(
  `2007` = "2007", `2008` = "2008", `2009` = "2009",
  `2010` = "2010", `2011` = "2011", `2012` = "2012",
  `2015` = "2015", `2018` = "2018", `2019` = "2019",
  `2020` = "2020", `2021` = "2021", `2022` = "2022"
) |>
# Imposing the FiveThirtyEight (538) theme on the `gt` table.
gt_theme_538() |>
# Styling the title cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_title()
) |>
# Styling the column labels cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center"
    ),
```

```
    cell_fill(
    color = "#D6D6D6"
    )
  ),
  locations = cells_column_labels()
) |>
# Styling the row grouping to make them prettier.
tab_style(
  style = list(
    cell_text(
      style = "italic",
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_row_groups()
) |>
# Styling the body cells to make them prettier.
tab_style(
  style = cell_text(align = "center"),
  locations = cells_body()
) |>
# Setting other general table options for prettier visualisation.
tab_options(
  table.font.size = px(14),
  heading.align = "center",
  row_group.font.weight = "bold",
  heading.subtitle.font.size = px(16),
  heading.title.font.size = px(18),
  column_labels.hidden = FALSE
) |>
# Setting the source note to comply with editorial requirements.
tab_source_note(
  source_note = "Note: Only employed respondents.\nSource: Own
  ↪  calculations, EU-SILC 2007-22 (N = 1.296.594); weighted sample."
) |>
# Styling the source notes cells to make them prettier.
tab_style(
  style = cell_text(align = "center"),
  locations = cells_source_notes()
```

```
  )

# Saving the `gt` table as a `.png` file.
gtsave(tablea24_gt, "Visuals/Tables/TableA24.png")
```

Table A2.4. Yearly Trends in Average Weekly Working Hours

by Sex and Migrant Background

2007	2008	2009	2010	2011	2012	2015	2018	2019	2020	2021	2022
Non-EU Migrant Women											
35.69h	35.27h	35.14h	34.79h	34.35h	33.95h	34.42h	33.64h	34.07h	33.42h	33.99h	34.50h
EU Migrant Women											
32.39h	33.60h	34.16h	33.96h	33.47h	33.32h	33.80h	34.86h	34.94h	34.40h	35.00h	35.63h
Native Women											
35.64h	35.54h	35.41h	35.52h	35.59h	35.75h	35.87h	35.94h	36.12h	36.24h	36.41h	36.51h
Non-EU Migrant Men											
41.80h	41.68h	40.85h	39.55h	39.70h	39.57h	39.14h	39.21h	39.05h	39.11h	38.43h	39.10h
EU Migrant Men											
41.60h	42.03h	41.45h	41.23h	41.69h	40.51h	41.56h	41.06h	40.67h	39.91h	40.07h	40.57h
Native Men											
40.85h	40.69h	40.39h	40.32h	40.31h	40.35h	40.27h	40.21h	40.05h	39.94h	39.90h	39.90h

Note: Only employed respondents. Source: Own calculations, EU-SILC 2007-22 (N = 1.296.594); weighted sample.

Figure 2.2 is appropriately styled and saved as a `.svg` file using the `ggplot2` package.

```r
# Figure 2.2. Yearly Trends in Average Weekly Working Hours by Sex and
 ↪ Migrant Background.

figure22 <- tablea24 |>
  # Recoding all years as numeric in a mutate() call, so that `ggplot2` can
   ↪ generate a line plot.
  mutate(year = as.numeric(year)) |>
  # Generating a lineplot of average weekly working hours by year, sex, and
   ↪ migrant background.
  ggplot(aes(x = year, y = PL060, linetype = sexmig, color = sexmig)) +
  # Setting an appropriate line width to facilitate the figure's
   ↪ interpretability.
  geom_line(linewidth = 0.7) +
  # Manually annotating rectangles of differing grey shades to highlight the
   ↪ Great Recession and the COVID-19 pandemic.
  annotate("rect", xmin = 2, xmax = 3, ymin = -Inf, ymax = Inf,
           alpha = 0.3, fill = "#A3A3A3") +
  annotate("rect", xmin = 3, xmax = 6, ymin = -Inf, ymax = Inf,
           alpha = 0.3, fill = "#C2C2C2") +
  annotate("rect", xmin = 10, xmax = 11, ymin = -Inf, ymax = Inf,
           alpha = 0.3, fill = "#A3A3A3") +
  annotate("rect", xmin = 11, xmax = 12, ymin = -Inf, ymax = Inf,
           alpha = 0.3, fill = "#C2C2C2") +
  # Assigning appropriate labels to all rectangles to highlight the Great
   ↪ Recession and the COVID-19 pandemic.
  geom_richtext(data = tibble(
    year = c(2.5, 4.5, 10.5, 11.5),
    perc = c(43.5, 43.5, 43.5, 43.5),
    label = c("Great Recession<br>(short-term)", "Great
     ↪ Recession<br>(long-term)",
              "COVID-19<br>(short-term)", "COVID-19<br>(long-term)")
    ),
    aes(x = year, y = perc, label = label),
    inherit.aes = FALSE, size = 3.5, color = "black", fontface = "bold") +
  # Setting a continuous year-by-year scale on the x-axis.
  scale_x_continuous(breaks = c(1:12),
                     labels = c("2007", "2008", "2009", "2010", "2011",
                      ↪ "2012",
                                 "2015", "2018", "2019", "2020", "2021",
                                  ↪ "2022")
                     ) +
  # Setting a continuous hour-based scale on the y-axis.
```

```
  scale_y_continuous(breaks = seq(0, 70, 1), limits = c(32, 44)) +
  # Setting combinations of grey shades and line types to highlight sex and
   ↪  migrant background categories.
  scale_linetype_manual(values = c("solid", "dotted", "twodash",
                                   "solid", "dotted", "twodash")) +
  scale_color_manual(values = c("#000000", "#000000", "#000000",
                                "#999999", "#999999", "#999999")) +
  # Styling axes and legends to make the figure look prettier.
  guides(color = guide_legend(byrow = TRUE), linetype = guide_legend(byrow =
   ↪  TRUE)) +
  labs(
    title = "",
    subtitle = "",
    x = "",
    y = "Average Weekly Working Hours",
    color = "",
    linetype = ""
  ) +
  # Styling the overall theme to make the figure look prettier.
  theme_bw(base_size = 14) +
  theme(
    plot.title.position = "plot",
    plot.title = element_text(hjust = 0.5, face = "bold", size = 18),
    plot.subtitle = element_text(hjust = 0.5, face = "italic", size = 14),
    legend.position = "bottom",
    strip.text = element_text(face = "bold")
  )

# Saving the `ggplot2` figure as a `.svg` file.
ggsave(filename = "Visuals/Figures/Figure22.svg",
       plot = figure22, width = 12, height = 8, dpi = 600)
```

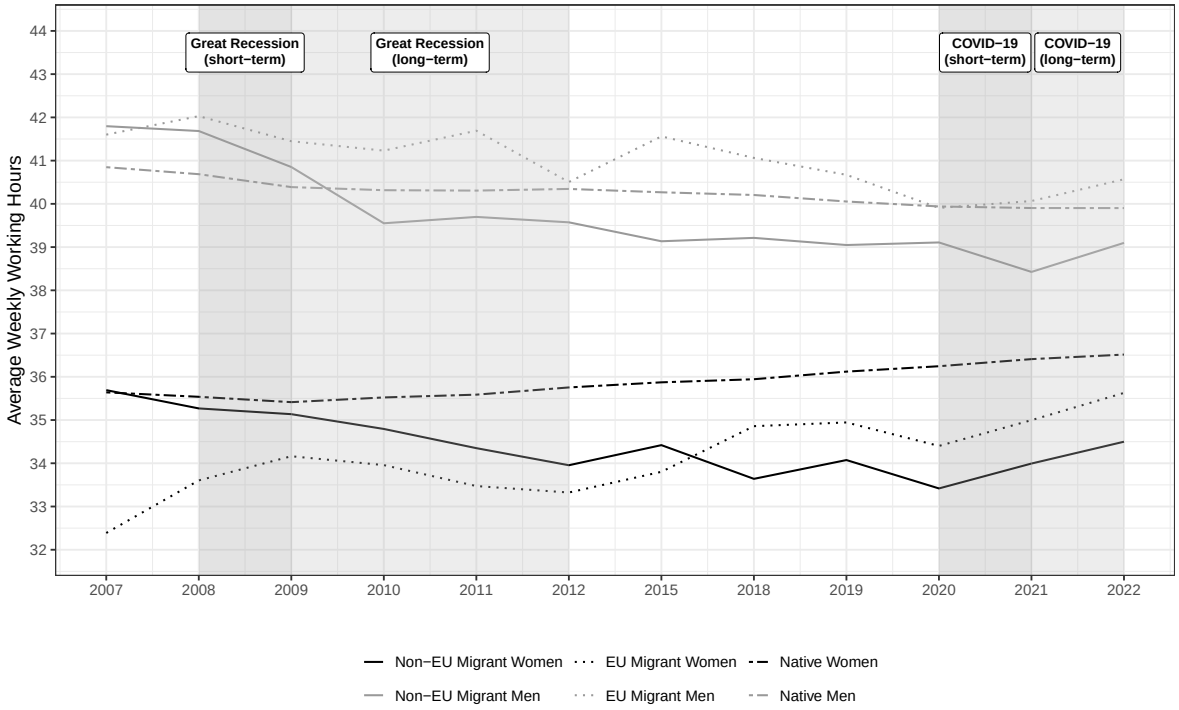

Figure 2.2. *Yearly Trends in Average Weekly Working Hours by Sex and Migrant Background.*

3.3 Table A2.5 / Figure 2.3

Figure 2.3 derives from the raw format tibble that synthesises the information contained in **Table A2.5**.

```
tablea25 <- d_emp |>
  # Recoding all sex and migrant background categories to enhance their
   ↪  readability, using fct_recode() in a mutate() call.
  mutate(
    sexmig = fct_recode(sexmig,
                        "Non-EU Migrant Women" = "NonEUMigWo",
                        "EU Migrant Women" = "EUMigWo",
                        "Native Women" = "NatWo",
                        "Non-EU Migrant Men" = "NonEUMigMe",
                        "EU Migrant Men" = "EUMigMe",
                        "Native Men" = "NatMe")) |>
```

```
  # Applying design weights with the as_survey() function from the `srvyr`
   ↪ package.
  as_survey(weights = RB050) |>
  # Grouping by year, sex and migrant background, and temporary or permanent
   ↪ employment categories.
  group_by(year, sexmig, temp) |>
  # Counting the weighted number of respondents within each group.
  summarise(weighted_n = survey_total()) |>
  # Transforming the weighted counts into percentages and rounding at two
   ↪ decimal places.
  mutate(perc = round(weighted_n / sum(weighted_n) * 100, 2)) |>
  # Resetting the grouping.
  ungroup() |>
  # Removing the weighted counts and their associated standard errors.
  select(-weighted_n, -weighted_n_se)
```

Table A2.5 is appropriately styled and saved as a `.png` file using the `gt` package.

```
# Table A2.5. Yearly Trends in Rates of Temporary and Permanent Employment,
 ↪ by Sex and Migrant Background.

tablea25_gt <- tablea25 |>
  # Recoding all percentages and years as strings in a mutate() call, so that
   ↪ `gt` does not mess with their visualisation.
  mutate(perc = sprintf("%.2f%%", perc),
         year = as.character(year)) |>
  # Pivoting the table to a wider format, assigning a separate column of
   ↪ percentages to each year.
  pivot_wider(names_from = year, values_from = perc) |>
  # Building the `gt` table, setting row names as temporary or permanent
   ↪ employment categories, and grouping rows by sex and migrant background.
  gt(rowname_col = "temp", groupname_col = "sexmig") |>
  # Assigning an appropriate title and subtitle to the `gt` table.
  tab_header(
    title = md("**Table A2.5. Yearly Trends in Rates of Temporary and
     ↪ Permanent Employment**"),
    subtitle = md("*by Sex and Migrant Background*")
  ) |>
  # Setting column labels as strings, so that `gt` does not mess with their
   ↪ visualisation.
  cols_label(
    `2007` = "2007", `2008` = "2008", `2009` = "2009",
```

```
  `2010` = "2010", `2011` = "2011", `2012` = "2012",
  `2015` = "2015", `2018` = "2018", `2019` = "2019",
  `2020` = "2020", `2021` = "2021", `2022` = "2022"
) |>
# Imposing the FiveThirtyEight (538) theme on the `gt` table.
gt_theme_538() |>
# Fixing the stubhead label to make it clear that each column represents a
 ↪ different year.
tab_stubhead(label = "Year") |>
# Styling the stubhead cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_stubhead()
  ) |>
# Styling the title cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_title()
) |>
# Styling the column labels cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
```

```
      style = "italic",
      align = "center"
    ),

    cell_fill(
    color = "#D6D6D6"
    )
  ),
  locations = cells_column_labels()
) |>
# Styling the row grouping cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      style = "italic",
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_row_groups()
) |>
# Styling the body cells to make them prettier.
tab_style(
  style = cell_text(align = "center"),
  locations = cells_body()
) |>
# Setting other general table options for prettier visualisation.
tab_options(
  table.font.size = px(14),
  heading.align = "center",
  row_group.font.weight = "bold",
  heading.subtitle.font.size = px(16),
  heading.title.font.size = px(18),
  column_labels.hidden = FALSE
) |>
# Setting the source note to comply with editorial requirements.
tab_source_note(
  source_note = "Note: Only employed respondents.\nSource: Own
  ↪  calculations, EU-SILC 2007-22 (N = 1.296.594); weighted sample."
) |>
```

```
  # Styling the source notes cells to make them prettier.
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_source_notes()
  )

# Saving the `gt` table as a `.png` file.
gtsave(tablea25_gt, "Visuals/Tables/TableA25.png")
```

Table A2.5. Yearly Trends in Rates of Temporary and Permanent Employment

by Sex and Migrant Background

Year	2007	2008	2009	2010	2011	2012	2015	2018	2019	2020	2021	2022
Non-EU Migrant Women												
Permanent job	59.76%	60.42%	71.26%	71.89%	70.41%	66.17%	69.74%	68.72%	64.27%	72.82%	74.78%	76.83%
Temporary job	40.24%	39.58%	28.74%	28.11%	29.59%	33.83%	30.26%	31.28%	35.73%	27.18%	25.22%	23.17%
EU Migrant Women												
Permanent job	81.24%	75.62%	79.50%	83.28%	80.24%	81.63%	77.02%	78.18%	80.48%	85.14%	85.53%	85.25%
Temporary job	18.76%	24.38%	20.50%	16.72%	19.76%	18.37%	22.98%	21.82%	19.52%	14.86%	14.47%	14.75%
Native Women												
Permanent job	81.56%	82.48%	83.83%	83.52%	83.70%	83.77%	82.41%	83.66%	84.38%	85.61%	83.90%	85.22%
Temporary job	18.44%	17.52%	16.17%	16.48%	16.30%	16.23%	17.59%	16.34%	15.62%	14.39%	16.10%	14.78%
Non-EU Migrant Men												
Permanent job	60.45%	63.26%	68.82%	71.24%	67.91%	68.63%	67.69%	68.46%	66.44%	72.78%	75.17%	82.05%
Temporary job	39.55%	36.74%	31.18%	28.76%	32.09%	31.37%	32.31%	31.54%	33.56%	27.22%	24.83%	17.95%
EU Migrant Men												
Permanent job	85.86%	75.48%	78.54%	80.99%	81.13%	81.27%	86.12%	85.96%	84.93%	87.86%	87.59%	90.97%
Temporary job	14.14%	24.52%	21.46%	19.01%	18.87%	18.73%	13.88%	14.04%	15.07%	12.14%	12.41%	9.03%
Native Men												
Permanent job	83.27%	84.76%	86.42%	85.50%	85.29%	85.12%	83.79%	85.19%	86.25%	87.47%	85.12%	86.77%
Temporary job	16.73%	15.24%	13.58%	14.50%	14.71%	14.88%	16.21%	14.81%	13.75%	12.53%	14.88%	13.23%

Note: Only employed respondents. Source: Own calculations, EU-SILC 2007-22 (N = 1.296.594); weighted sample.

Figure 2.3 is appropriately styled and saved as a `.svg` file using the `ggplot2` package.

```
# Figure 2.3. Yearly Trends in Temporary Employment Rates by Sex and Migrant
 ↪ Background.
```

```r
figure23 <- tablea25 |>
  # Recoding all years as numeric in a mutate() call, so that `ggplot2` can
   ↪ generate a line plot.
  mutate(year = as.numeric(year),
         # Recoding all sex and migrant background categories to enhance
          ↪ their readability.
         sexmig = factor(sexmig, levels = c(
        "Non-EU Migrant Women", "EU Migrant Women", "Native Women",
        "Non-EU Migrant Men", "EU Migrant Men", "Native Men"))) |>
  # Keeping only respondents in temporary employment using the filter()
   ↪ `tidyverse` function.
  filter(temp %in% c("Temporary job")) |>
  # Generating a lineplot of temporary employment rates by year, sex, and
   ↪ migrant background.
  ggplot(aes(x = year, y = perc, linetype = sexmig, color = sexmig)) +
  # Setting an appropriate line width to facilitate the figure's
   ↪ interpretability.
  geom_line(linewidth = 0.7) +
  # Manually annotating rectangles of differing grey shades to highlight the
   ↪ Great Recession and the COVID-19 pandemic.
  annotate("rect", xmin = 2, xmax = 3, ymin = -Inf, ymax = Inf,
           alpha = 0.3, fill = "#A3A3A3") +
  annotate("rect", xmin = 3, xmax = 6, ymin = -Inf, ymax = Inf,
           alpha = 0.3, fill = "#C2C2C2") +
  annotate("rect", xmin = 10, xmax = 11, ymin = -Inf, ymax = Inf,
           alpha = 0.3, fill = "#A3A3A3") +
  annotate("rect", xmin = 11, xmax = 12, ymin = -Inf, ymax = Inf,
           alpha = 0.3, fill = "#C2C2C2") +
  # Assigning appropriate labels to all rectangles to highlight the Great
   ↪ Recession and the COVID-19 pandemic.
  geom_richtext(data = tibble(
    year = c(2.5, 4.5, 10.5, 11.5),
    perc = c(41.5, 41.5, 41.5, 41.5),
    label = c("Great Recession<br>(short-term)", "Great
     ↪ Recession<br>(long-term)",
              "COVID-19<br>(short-term)", "COVID-19<br>(long-term)")
    ),
    aes(x = year, y = perc, label = label),
    inherit.aes = FALSE, size = 3.5, color = "black", fontface = "bold") +
  # Setting a continuous year-by-year scale on the x-axis.
  scale_x_continuous(breaks = c(1:12),
                     labels = c("2007", "2008", "2009", "2010", "2011",
                      ↪ "2012",
```

```
                                "2015", "2018", "2019", "2020", "2021",
                                ↪  "2022")
                    ) +
  # Setting a continuous percentage scale on the y-axis.
  scale_y_continuous(breaks = seq(0, 45, 1)) +
  # Setting combinations of grey shades and line types to highlight sex and
   ↪  migrant background categories.
  scale_linetype_manual(values = c("solid", "dotted", "twodash",
                                   "solid", "dotted", "twodash")) +
  scale_color_manual(values = c("#000000", "#000000", "#000000",
                                "#999999", "#999999", "#999999")) +
  # Styling axes and legends to make the figure look prettier.
  guides(color = guide_legend(byrow = TRUE), linetype = guide_legend(byrow =
   ↪  TRUE)) +
  labs(
    title = "",
    subtitle = "",
    x = "",
    y = "% of Respondents",
    color = "",
    linetype = ""
  ) +
  # Styling the overall theme to make the figure look prettier.
  theme_bw(base_size = 14) +
  theme(
    plot.title.position = "plot",
    plot.title = element_text(hjust = 0.5, face = "bold", size = 18),
    plot.subtitle = element_text(hjust = 0.5, face = "italic", size = 14),
    legend.position = "bottom",
    strip.text = element_text(face = "bold")
  )

# Saving the `ggplot2` figure as a `.svg` file.
ggsave(filename = "Visuals/Figures/Figure23.svg",
       plot = figure23, width = 12, height = 12, dpi = 600)
```

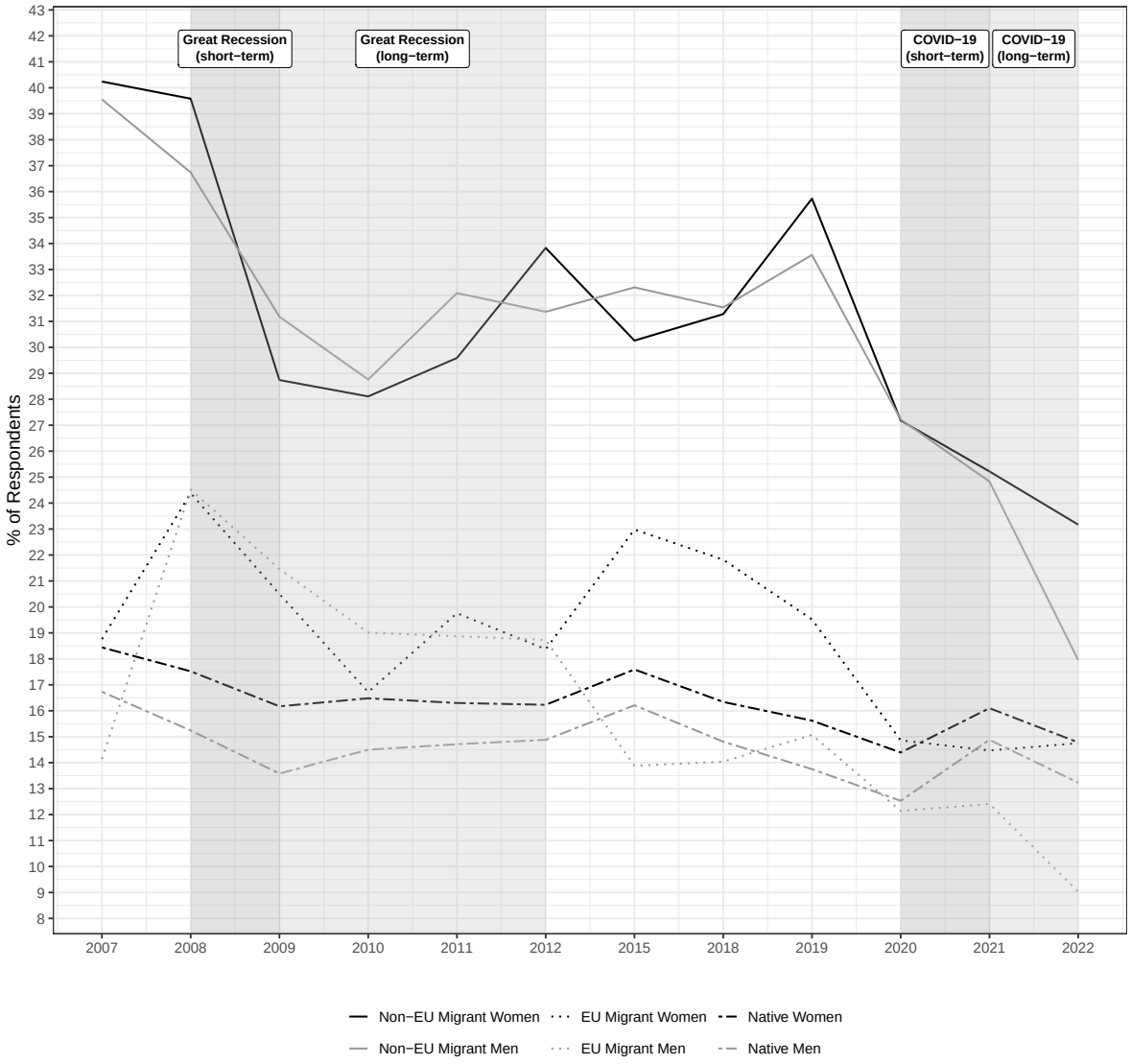

Figure 2.3. *Yearly Trends in Temporary Employment Rates by Sex and Migrant Background.*

3.4 Table A2.3

Table A2.3 derives from the raw format tibbles that synthesise the information contained in **Tables A2.4** and **A2.5**.

Table A2.3 is appropriately styled and saved as a `.png` file using the `gt` package.

```r
# Table A2.3. Changes in Weekly Working Hours and Rates of Temporary
 ↪  Employment, by Sex and Migrant Background.

tablea23_gt <- tablea24 |>
  # Isolating all years delimiting the Great Recession and the COVID-19
   ↪  pandemic, using the filter() `tidyverse` function.
  filter(year %in% c("2007", "2009", "2012", "2019", "2020", "2022")) |>
  # Pivoting the table to a wider format, assigning a separate column of
   ↪  average weekly working hours to each year.
  pivot_wider(names_from = year, values_from = PL060) |>
  # Calculating all the short-term and long-term percentage changes.
  mutate(
    change_07_09 = (`2009` - `2007`), # Great Recession (short-term).
    change_07_12 = (`2012` - `2007`), # Great Recession (long-term).
    change_19_20 = (`2020` - `2019`), # COVID-19 pandemic (short-term).
    change_19_22 = (`2022` - `2019`) # COVID-19 pandemic (long-term).
  ) |>
  # Pivoting the table to a wider format, assigning a single column to all
   ↪  percentage changes.
  pivot_longer(cols = c(change_07_09, change_07_12,
                        change_19_20, change_19_22),
               names_to = "period",
               values_to = "change") |>
  # Dropping all superfluous variables, using the select() `tidyverse`
   ↪  function.
  select(sexmig, period, change) |>
  # Recoding all variable labels to enhance their readability, using
   ↪  fct_recode() in a mutate() call.
  mutate(
    period = fct_recode(as.factor(period),
                        "Great Recession\n(2007-09, short-term)" =
                          "change_07_09",
                        "Great Recession\n(2007-12, long-term)" =
                          "change_07_12",
                        "COVID-19\n(2019-20, short-term)" =
                          "change_19_20",
                        "COVID-19\n(2019-22, long-term)" =
                          "change_19_22")) |>
  # Recoding all percentage changes as strings, so that `gt` does not mess
   ↪  with their visualisation.
  mutate(change = if_else(change > 0,
                          sprintf("+%.2fh", change),
```

```
                        sprintf("%.2fh", change))) |>
# Pivoting the table to a wider format, assigning a separate column of
 ↪  percentage changes to each time period category.
pivot_wider(names_from = period, values_from = change) |>
# Assigning a parameter identifier to distinguish average weekly working
 ↪  hours from rates of temporary employment.
mutate(parameter = "Weekly Working Hours") |>
# Binding the raw tibble for changes in average weekly working hours with
 ↪  the raw tibble for changes in rates of temporary employment.
rbind(tablea25 |>
        # Isolating all years delimiting the Great Recession and the
         ↪  COVID-19 pandemic, using the filter() `tidyverse` function.
        filter(year %in% c("2007", "2009", "2012", "2019", "2020", "2022"))
         ↪  |>
        # Pivoting the table to a wider format, assigning a separate column
         ↪  of percentages to each year.
        pivot_wider(names_from = year, values_from = perc) |>
        # Calculating all the short-term and long-term percentage changes.
        mutate(
          change_07_09 = (`2009` - `2007`), # Great Recession (short-term).
          change_07_12 = (`2012` - `2007`), # Great Recession (long-term).
          change_19_20 = (`2020` - `2019`), # COVID-19 pandemic
           ↪  (short-term).
          change_19_22 = (`2022` - `2019`) # COVID-19 pandemic (long-term).
          ) |>
        # Pivoting the table to a wider format, assigning a single column
         ↪  to all percentage changes.
        pivot_longer(cols = c(change_07_09, change_07_12,
                              change_19_20, change_19_22),
                     names_to = "period",
                     values_to = "change") |>
        # Dropping all superfluous variables, using the select()
         ↪  `tidyverse` function.
        select(sexmig, temp, period, change) |>
        # Recoding all variable labels to enhance their readability, using
         ↪  fct_recode() in a mutate() call.
        mutate(period = fct_recode(as.factor(period),
                     "Great Recession\n(2007-09, short-term)" =
                       "change_07_09",
                      "Great Recession\n(2007-12, long-term)" =
                       "change_07_12",
                      "COVID-19\n(2019-20, short-term)" =
```

```
                        "change_19_20",
                      "COVID-19\n(2019-22, long-term)" =
                        "change_19_22")) |>
        # Recoding all percentage changes as strings, so that `gt` does not
         ↪ mess with their visualisation.
        mutate(change = if_else(change > 0,
                                sprintf("+%.2f%%", change),
                                sprintf("%.2f%%", change))) |>
        # Keeping only changes in temporary employment, using the filter()
         ↪ `tidyverse` function.
        filter(temp %in% "Temporary job") |>
        # Dropping the now superfluous temporary employment identifier,
         ↪ using the select() `tidyverse` function.
        select(-temp) |>
        # Pivoting the table to a wider format, assigning a separate column
         ↪ of percentage changes to each time period category.
        pivot_wider(names_from = period, values_from = change) |>
        # Assigning a parameter identifier to distinguish average weekly
         ↪ working hours from rates of temporary employment.
        mutate(parameter = "Temporary Employment Rates")) |>
# Building the `gt` table, setting row names as sex and migrant background
 ↪ categories, and grouping rows by parameter.
gt(rowname_col = "sexmig", groupname_col = "parameter") |>
# Assigning an appropriate title and subtitle to the `gt` table.
tab_header(
  title = md("**Table A2.3. Changes in Weekly Working Hours and Temporary
   ↪ Employment Rates**"),
  subtitle = md("*by Sex and Migrant Background*")
) |>
# Imposing the FiveThirtyEight (538) theme on the `gt` table.
gt_theme_538() |>
# Fixing the stubhead label to make it clear that each column represents a
 ↪ different time period
tab_stubhead(label = "Time Period") |>
# Styling the stubhead cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle"
```

```
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_stubhead()
  ) |>
# Styling the title cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_title()
) |>
# Styling the column labels cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle",
      whitespace = "pre-wrap",
      transform = "capitalize"
    ),
    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_column_labels()
) |>
# Styling the row grouping cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      style = "italic",
```

```
        align = "center"
      ),

      cell_fill(
        color = "#D6D6D6"
      )
    ),
    locations = cells_row_groups()
  ) |>
  # Styling the body cells to make them prettier.
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_body()
  ) |>
  # Setting other general table options for prettier visualisation.
  tab_options(
    table.font.size = px(14),
    heading.align = "center",
    row_group.font.weight = "bold",
    heading.subtitle.font.size = px(16),
    heading.title.font.size = px(18)
  ) |>
  # Setting the source note to comply with editorial requirements.
  tab_source_note(
    source_note = "Note: Only employed respondents.\nChanges are calculated
    ↪  by subtracting the % in 2009/12/20/22 from the % in 2007/19.\nSource:
    ↪  Own calculations, EU-SILC 2007-22 (N = 1.296.594); weighted sample."
  ) |>
  # Styling the source notes cells to make them prettier.
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_source_notes()
  )

# Saving the `gt` table as a `.png` file.
gtsave(tablea23_gt, "Visuals/Tables/TableA23.png")
```

Table A2.3. Changes in Weekly Working Hours and Temporary Employment Rates

by Sex and Migrant Background

Time Period	*Great Recession (2007-09, Short-Term)*	*Great Recession (2007-12, Long-Term)*	*COVID-19 (2019-20, Short-Term)*	*COVID-19 (2019-22, Long-Term)*
		Weekly Working Hours		
Non-EU Migrant Women	-0.56h	-1.74h	-0.66h	+0.42h
EU Migrant Women	+1.77h	+0.94h	-0.54h	+0.68h
Native Women	-0.23h	+0.11h	+0.12h	+0.39h
Non-EU Migrant Men	-0.94h	-2.22h	+0.06h	+0.05h
EU Migrant Men	-0.15h	-1.10h	-0.77h	-0.10h
Native Men	-0.46h	-0.50h	-0.11h	-0.15h
		Temporary Employment Rates		
Non-EU Migrant Women	-11.50%	-6.41%	-8.55%	-12.56%
EU Migrant Women	+1.74%	-0.39%	-4.66%	-4.77%
Native Women	-2.27%	-2.21%	-1.23%	-0.84%
Non-EU Migrant Men	-8.37%	-8.18%	-6.34%	-15.61%
EU Migrant Men	+7.32%	+4.59%	-2.93%	-6.04%
Native Men	-3.15%	-1.85%	-1.22%	-0.52%

Note: Only employed respondents. Changes are calculated by subtracting the % in 2009/12/20/22 from the % in 2007/19. Source: Own calculations, EU-SILC 2007-22 (N = 1.296.594); weighted sample.

3.5 Table A2.6

Since the pipeline needed to generate the **Mutual Information Index of Occupational Segregation** (**M**) is computationally intensive, I exploit *parallel processing* to speed up the calculations, using the `future` package.

```
# Opening the parallel processing session, utilising all the 14 cores
 ↪  available on my machine.
plan(multisession, workers = 14)

# Binding the raw tibble containing ISCO-88 codes (2007-10) with the raw
 ↪  tibble including ISCO-08 codes (2011-2022).
tablea26 <- bind_rows(
  # Dropping all superfluous variables, using the select() `tidyverse`
   ↪  function.
  d |> select(year, sexmig, PL050_fct, RB050) |>
    # Keeping ISCO-88 codes only (2007-10), using the filter() `tidyverse`
     ↪  function.
    filter(year %in% c("2007", "2008", "2009", "2010")) |>
```

```
  # Splitting sex and migrant background categories in a mutate() call, to
   ↪ compute their separate as well as joint contribution to M.
  mutate(sex = fct_collapse(sexmig,
                            "Female" = c("NonEUMigWo", "EUMigWo", "NatWo"),
                            "Male"   = c("NonEUMigMe", "EUMigMe",
                             ↪ "NatMe")),
         mig = fct_collapse(sexmig,
                            "Non-EU Migrant" = c("NonEUMigWo",
                             ↪ "NonEUMigMe"),
                            "EU Migrant" = c("EUMigWo", "EUMigMe"),
                            "Native" = c("NatWo", "NatMe"))) |>
  # Dropping the superfluous joint sex and migrant background variable,
   ↪ using the select() `tidyverse` function.
  select(-sexmig) |>
  # Applying design weights with the as_survey() function from the `srvyr`
   ↪ package.
  as_survey(weights = RB050) |>
  # Grouping by year, sex, migrant background, and ISCO-88 codes.
  group_by(year, sex, mig, PL050_fct) |>
  # Counting the weighted number of respondents within each group.
  survey_count(name = "weighted_n") |>
  # Resetting the grouping.
  ungroup() |>
  # Removing the standard errors associated to the weighted counts.
  select(-weighted_n_se) |>
  # Transforming the tibble in the `data.table` format, using the
   ↪ prepare_data() `mutualinf` function.
  prepare_data(vars = "all_vars", fw = "weighted_n") |>
  # Computing M, using the mutual() `mutualinf` function.
  mutual(
    group = c("sex", "mig"),
    unit = c("PL050_fct"),
    by = "year",
    contribution.from = "group_vars"
    ),

# Dropping all superfluous variables, using the select() `tidyverse`
 ↪ function.
d |> select(year, sexmig, PL051_fct, RB050) |>
  # Keeping ISCO-08 codes only (2011-22), using the filter() `tidyverse`
   ↪ function.
  filter(!(year %in% c("2007", "2008", "2009", "2010"))) |>
```

```
    # Splitting sex and migrant background categories in a mutate() call, to
     ↪  compute their separate as well as joint contribution to M.
    mutate(sex = fct_collapse(sexmig,
                              "Female" = c("NonEUMigWo", "EUMigWo", "NatWo"),
                              "Male"   = c("NonEUMigMe", "EUMigMe",
                               ↪  "NatMe")),
           mig = fct_collapse(sexmig,
                              "Non-EU Migrant" = c("NonEUMigWo",
                               ↪  "NonEUMigMe"),
                              "EU Migrant" = c("EUMigWo", "EUMigMe"),
                              "Native" = c("NatWo", "NatMe"))) |>
    # Dropping the superfluous joint sex and migrant background variable,
     ↪  using the select() `tidyverse` function.
    select(-sexmig) |>
    # Applying design weights with the as_survey() function from the `srvyr`
     ↪  package.
    as_survey(weights = RB050) |>
    # Grouping by year, sex, migrant background, and ISCO-08 codes.
    group_by(year, sex, mig, PL051_fct) |>
    # Counting the weighted number of respondents within each group.
    survey_count(name = "weighted_n") |>
    # Resetting the grouping.
    ungroup() |>
    # Removing the standard errors associated to the weighted counts.
    select(-weighted_n_se) |>
    # Transforming the tibble in the `data.table` format, using the
     ↪  prepare_data() `mutualinf` function.
    prepare_data(vars = "all_vars", fw = "weighted_n") |>
    # Computing M, using the mutual() `mutualinf` function.
    mutual(
      group = c("sex", "mig"),
      unit = c("PL051_fct"),
      by = "year",
      contribution.from = "group_vars"
      ))

# Closing the parallel processing session.
plan(sequential)
```

Table A2.6 is appropriately styled and saved as a `.png` file using the `gt` package.

```
# Table A2.6. Yearly Trends in Mutual Information Index of Occupational
 ↪  Segregation, by Sex and Migrant Background.

tablea26_gt <- tablea26 |>
  # Recoding all figures and years as strings in a mutate() call, so that
   ↪  `gt` does not mess with their visualisation.
  mutate(M = sprintf("%.5f", M),
         C_sex = sprintf("%.5f", C_sex),
         C_mig = sprintf("%.5f", C_mig),
         interaction = sprintf("%.5f", interaction),
         year = as.character(year)) |>
  # Building the `gt` table, setting row names as years,
  gt(rowname_col = "year") |>
  # Assigning an appropriate title and subtitle to the `gt` table.
  tab_header(
    title = md("**Table A2.6. Yearly Trends in Mutual Information Index of
     ↪  Occupational Segregation**"),
    subtitle = md("*by Sex and Migrant Background*")
  ) |>
  # Renaming column labels to foster their readability.
  cols_label(
    `M` = "Mutual Information Index\nof Occupational Segregation",
    `C_sex` = "Contribution\nfrom Sex",
    `C_mig` = "Contribution from\nMigrant Background",
    `interaction` = "Contribution\nfrom Interaction"
  ) |>
  # Imposing the FiveThirtyEight (538) theme on the `gt` table.
  gt_theme_538() |>
  # Fixing the stubhead label to make it clear that each column represents a
   ↪  different year.
  tab_stubhead(label = "Year") |>
  # Styling the stubhead cells to make them prettier.
  tab_style(
    style = list(
      cell_text(
        weight = "bold",
        style = "italic",
        align = "center",
        v_align = "middle"
      ),

      cell_fill(
        color = "#D6D6D6"
```

```
    )
  ),
  locations = cells_stubhead()
  ) |>
# Styling the title cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_title()
) |>
# Styling the column labels cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle",
      whitespace = "pre-wrap",
      transform = "capitalize"
    ),

    cell_fill(
    color = "#D6D6D6"
    )
  ),
  locations = cells_column_labels()
) |>
# Styling the row grouping cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      style = "italic",
      align = "center"
    ),
```

```
      cell_fill(
        color = "#D6D6D6"
      )
    ),
    locations = cells_row_groups()
  ) |>
  # Styling the body cells to make them prettier.
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_body()
  ) |>
  # Setting other general table options for prettier visualisation.
  tab_options(
    table.font.size = px(14),
    heading.align = "center",
    row_group.font.weight = "bold",
    heading.subtitle.font.size = px(16),
    heading.title.font.size = px(18),
    column_labels.hidden = FALSE
  ) |>
  # Setting the source note to comply with editorial requirements.
  tab_source_note(
    source_note = "Note: Occupations are classified using 2-digit ISCO-88
    ↪  (2007-10) or ISCO-08 (2011-22).\nSource: Own calculations, EU-SILC
    ↪  2007-22 (N = 1.838.684); weighted sample."
  ) |>
  # Styling the source notes cells to make them prettier.
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_source_notes()
  )

# Saving the `gt` table as a `.png` file.
gtsave(tablea26_gt, "Visuals/Tables/TableA26.png")
```

Table A2.6. Yearly Trends in Mutual Information Index of Occupational Segregation

by Sex and Migrant Background

Year	Mutual Information Index Of Occupational Segregation	Contribution From Sex	Contribution From Migrant Background	Contribution From Interaction
2007	0.17877	0.16271	0.01734	-0.00128
2008	0.17794	0.16350	0.01567	-0.00124
2009	0.17637	0.16240	0.01521	-0.00124
2010	0.17230	0.15975	0.01357	-0.00102
2011	0.20015	0.18457	0.01690	-0.00132
2012	0.19254	0.17707	0.01705	-0.00159
2015	0.19231	0.17794	0.01537	-0.00100
2018	0.19021	0.17637	0.01539	-0.00155
2019	0.18388	0.17110	0.01457	-0.00180
2020	0.18251	0.16816	0.01577	-0.00142
2021	0.17604	0.16156	0.01581	-0.00133
2022	0.17233	0.15671	0.01724	-0.00162

Note: Occupations are classified using 2-digit ISCO-88 (2007-10) or ISCO-08 (2011-22). Source: Own calculations, EU-SILC 2007-22 (N = 1.838.684); weighted sample.

3.6 Table A2.7a / Table A2.7b

Tables A2.7a and **A2.7b** derive from a raw format tibble that synthesises information on occupational concentration among both men and women.

```
tablea27 <- d |>
  # Dropping all superfluous variables, using the select() `tidyverse`
   ↪ function.
  select(year, sexmig, isco, RB050) |>
  # Applying design weights with the as_survey() function from the `srvyr`
   ↪ package.
  as_survey(weights = RB050) |>
  # Grouping by year, sex and migrant background, and occupational code.
  group_by(year, sexmig, isco) |>
  # Counting the weighted number of respondents within each group.
  summarise(weighted_n = survey_total()) |>
  # Computing occupational concentration percentages, rounded to two decimal
   ↪ places.
  mutate(perc = round(weighted_n / sum(weighted_n) * 100, 2)) |>
  # Resetting the grouping.
```

```
  ungroup() |>
  # Removing the weighted counts and their associated standard errors.
  select(-weighted_n, -weighted_n_se) |>
  # Recoding all sex and migrant background categories to enhance their
   ↪  readability, using fct_collapse() and fct_recode() in a mutate() call.
  mutate(
    sex = fct_collapse(sexmig,
                           "Women" = c("NonEUMigWo", "EUMigWo", "NatWo"),
                           "Men"   = c("NonEUMigMe", "EUMigMe", "NatMe")),

    mig = fct_collapse(sexmig,
                           "Non-EU Migrant" = c("NonEUMigWo", "NonEUMigMe"),
                           "EU Migrant" = c("EUMigWo", "EUMigMe"),
                           "Native" = c("NatWo", "NatMe")),

    sexmig = fct_recode(sexmig,
                        "Non-EU Migrant Women" = "NonEUMigWo",
                        "EU Migrant Women" = "EUMigWo",
                        "Native Women" = "NatWo",
                        "Non-EU Migrant Men" = "NonEUMigMe",
                        "EU Migrant Men" = "EUMigMe",
                        "Native Men" = "NatMe"))
```

Table A2.7a is appropriately styled and saved as a `.png` file using the `gt` package.

```
# Table A2.7a. Yearly Trends in Occupational Concentration (1-digit ISCO)
 ↪  among Women, by Migrant Background.

tablea27a_gt <- tablea27 |>
  # Recoding all percentages, years, and occupational categories as strings
   ↪  in a mutate() call, so that `gt` does not mess with their
   ↪  visualisation.
  mutate(perc = sprintf("%.2f%%", perc),
         year = as.character(year),
         isco = factor(isco,
                       levels = c("1", "2", "3", "4", "5", "6", "7", "8",
                        ↪  "9"),
                       labels = c("Managers", "Professionals", "Technicians",
                                  "Clerical workers", "Service workers",
                                  "Skilled workers",
                                  "Craft and related trades workers",
                                  "Operators", "Elementary workers"))) |>
```

```
# Keeping women only, using the filter() `tidyverse` function.
filter(sex == "Women") |>
# Dropping all superfluous variables, using the select() `tidyverse`
 ↪  function.
select(-sexmig, -sex) |>
# Pivoting the table to a wider format, assigning a separate column of
 ↪  percentages to each year.
pivot_wider(names_from = year, values_from = perc) |>
# Building the `gt` table, setting row names as occupational categories,
 ↪  and grouping rows by migrant background.
gt(rowname_col = "isco", groupname_col = "mig") |>
# Assigning an appropriate title and subtitle to the `gt` table.
tab_header(
  title = md("**Table A2.7a. Yearly Trends in Occupational Concentration
   ↪  (1-digit ISCO)**"),
  subtitle = md("*among Women, by Migrant Background*")
) |>
# Setting column labels as strings, so that `gt` does not mess with their
 ↪  visualisation.
cols_label(
  `2007` = "2007", `2008` = "2008", `2009` = "2009",
  `2010` = "2010", `2011` = "2011", `2012` = "2012",
  `2015` = "2015", `2018` = "2018", `2019` = "2019",
  `2020` = "2020", `2021` = "2021", `2022` = "2022"
) |>
# Imposing the FiveThirtyEight (538) theme on the `gt` table.
gt_theme_538() |>
# Fixing the stubhead label to make it clear that each column represents a
 ↪  different year.
tab_stubhead(label = "Year") |>
# Styling the stubhead cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle"
    ),

    cell_fill(
      color = "#D6D6D6"
```

```
    )
  ),
  locations = cells_stubhead()
  ) |>
# Styling the title cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_title()
) |>
# Styling the column labels cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle"
    ),

    cell_fill(
    color = "#D6D6D6"
    )
  ),
  locations = cells_column_labels()
) |>
# Styling the row grouping cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      style = "italic",
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
```

```
      )
    ),
    locations = cells_row_groups()
  ) |>
  # Styling the body cells to make them prettier.
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_body()
  ) |>
  # Setting other general table options for prettier visualisation.
  tab_options(
    table.font.size = px(14),
    heading.align = "center",
    row_group.font.weight = "bold",
    heading.subtitle.font.size = px(16),
    heading.title.font.size = px(18),
    column_labels.hidden = FALSE
  ) |>
  # Setting the source note to comply with editorial requirements.
  tab_source_note(
    source_note = "Note: Women only. Occupations are classified using 2-digit
    ↪  ISCO-88 (2007-10) or ISCO-08 (2011-22).\nSource: Own calculations,
    ↪  EU-SILC 2007-22 (N = 1.838.684); weighted sample."
  ) |>
  # Styling the source notes cells to make them prettier.
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_source_notes()
  )

# Saving the `gt` table as a `.png` file.
gtsave(tablea27a_gt, "Visuals/Tables/TableA27a.png")
```

Table A2.7a. Yearly Trends in Occupational Concentration (1-digit ISCO)

among Women, by Migrant Background

Year	*2007*	*2008*	*2009*	*2010*	*2011*	*2012*	*2015*	*2018*	*2019*	*2020*	*2021*	*2022*
						Non-EU Migrant						
Managers	1.16%	1.23%	1.93%	1.44%	1.41%	1.36%	1.18%	1.79%	1.54%	1.12%	1.85%	2.60%
Professionals	4.63%	4.61%	4.36%	5.29%	6.56%	8.19%	8.99%	9.91%	11.49%	12.07%	17.18%	16.83%
Technicians	3.40%	5.13%	6.00%	7.22%	5.79%	6.09%	5.13%	8.37%	8.05%	8.50%	11.06%	9.92%
Clerical workers	7.39%	7.27%	8.27%	8.69%	6.59%	6.33%	4.75%	7.35%	8.38%	8.62%	9.32%	10.31%
Service workers	26.97%	28.21%	23.28%	26.45%	26.10%	29.28%	29.88%	26.44%	27.21%	27.14%	24.16%	24.84%
Skilled workers	0.90%	1.02%	0.67%	1.14%	0.75%	1.76%	1.53%	1.57%	1.36%	1.50%	0.99%	0.72%
Craft and related trades workers	3.59%	3.57%	3.11%	3.31%	4.53%	3.56%	4.13%	2.48%	3.14%	2.80%	2.37%	2.07%
Operators	3.65%	4.40%	4.58%	4.77%	3.62%	2.14%	3.26%	4.03%	3.60%	2.93%	3.29%	3.07%
Elementary workers	48.30%	44.57%	47.80%	41.69%	44.66%	41.28%	41.15%	38.06%	35.23%	35.32%	29.77%	29.65%
						EU Migrant						
Managers	3.61%	3.64%	3.96%	3.78%	2.84%	3.63%	2.56%	3.45%	3.36%	2.98%	2.88%	2.87%
Professionals	14.12%	11.64%	10.30%	10.61%	11.26%	13.46%	15.48%	18.96%	16.23%	18.07%	22.57%	23.35%
Technicians	13.51%	13.53%	12.37%	13.13%	13.59%	11.61%	9.05%	9.91%	10.24%	9.92%	13.35%	12.10%
Clerical workers	14.97%	12.38%	11.78%	12.60%	10.10%	10.50%	10.46%	10.31%	11.02%	11.24%	13.82%	15.23%
Service workers	18.41%	22.51%	22.08%	21.56%	22.01%	22.52%	24.32%	21.53%	20.28%	21.18%	19.19%	19.12%
Skilled workers	0.93%	0.45%	0.88%	1.20%	0.95%	2.20%	1.43%	0.97%	0.91%	1.51%	1.15%	1.65%
Craft and related trades workers	2.92%	3.69%	4.64%	3.48%	2.73%	3.65%	2.32%	2.37%	2.79%	3.18%	2.88%	2.82%
Operators	1.82%	2.52%	3.18%	4.10%	4.58%	2.92%	2.68%	3.82%	5.94%	2.69%	2.70%	2.81%
Elementary workers	29.72%	29.65%	30.82%	29.54%	31.93%	29.49%	31.69%	28.67%	29.22%	29.23%	21.44%	20.05%
						Native						
Managers	2.98%	3.12%	3.31%	3.31%	2.93%	3.49%	2.80%	3.36%	3.13%	2.79%	3.35%	3.89%
Professionals	13.77%	14.24%	14.73%	15.13%	15.47%	18.32%	19.72%	21.10%	21.80%	23.49%	25.48%	25.76%
Technicians	16.27%	16.89%	17.25%	17.10%	16.28%	15.14%	15.33%	15.24%	15.37%	15.43%	16.04%	16.78%
Clerical workers	18.79%	18.38%	17.82%	17.53%	16.50%	15.89%	15.20%	14.81%	14.52%	14.74%	15.09%	15.29%
Service workers	21.77%	21.54%	21.73%	22.10%	23.60%	23.29%	23.31%	23.58%	23.58%	22.52%	21.40%	20.15%
Skilled workers	0.89%	0.89%	0.91%	0.87%	0.75%	1.00%	0.93%	0.70%	0.78%	0.79%	0.89%	0.85%
Craft and related trades workers	5.75%	5.20%	4.71%	4.38%	4.08%	4.59%	3.69%	3.35%	3.37%	3.30%	3.17%	3.06%
Operators	3.89%	4.01%	4.18%	4.47%	4.67%	3.67%	4.22%	4.02%	4.15%	3.90%	3.66%	3.61%
Elementary workers	15.89%	15.73%	15.37%	15.11%	15.72%	14.61%	14.79%	13.83%	13.32%	13.04%	10.91%	10.63%

Note: Women only. Occupations are classified using 2-digit ISCO-88 (2007-10) or ISCO-08 (2011-22). Source: Own calculations, EU-SILC 2007-22 (N = 1.838.684); weighted sample.

Table A2.7b is appropriately styled and saved as a `.png` file using the `gt` package.

```
# Table A2.7b. Yearly Trends in Occupational Concentration (1-digit ISCO)
 ↪  among Men, by Migrant Background.

tablea27b_gt <- tablea27 |>
  # Recoding all percentages, years, and occupational categories as strings
   ↪  in a mutate() call, so that `gt` does not mess with their
   ↪  visualisation.
  mutate(perc = sprintf("%.2f%%", perc),
         year = as.character(year),
         isco = factor(isco,
                       levels = c("1", "2", "3", "4", "5", "6", "7", "8",
                        ↪  "9"),
                       labels = c("Managers", "Professionals", "Technicians",
                                  "Clerical workers", "Service workers",
                                  "Skilled workers",
                                  "Craft and related trades workers",
                                  "Operators", "Elementary workers"))) |>
  # Keeping women only, using the filter() `tidyverse` function.
  filter(sex == "Men") |>
  # Dropping all superfluous variables, using the select() `tidyverse`
   ↪  function.
  select(-sexmig, -sex) |>
  # Pivoting the table to a wider format, assigning a separate column of
   ↪  percentages to each year.
  pivot_wider(names_from = year, values_from = perc) |>
  # Building the `gt` table, setting row names as occupational categories,
   ↪  and grouping rows by migrant background.
  gt(rowname_col = "isco", groupname_col = "mig") |>
  # Assigning an appropriate title and subtitle to the `gt` table.
  tab_header(
    title = md("**Table A2.7b. Yearly Trends in Occupational Concentration
     ↪  (1-digit ISCO)**"),
    subtitle = md("*among Men, by Migrant Background*")
  ) |>
  # Setting column labels as strings, so that `gt` does not mess with their
   ↪  visualisation.
  cols_label(
    `2007` = "2007", `2008` = "2008", `2009` = "2009",
    `2010` = "2010", `2011` = "2011", `2012` = "2012",
    `2015` = "2015", `2018` = "2018", `2019` = "2019",
    `2020` = "2020", `2021` = "2021", `2022` = "2022"
```

```
) |>
# Imposing the FiveThirtyEight (538) theme on the `gt` table.
gt_theme_538() |>
# Fixing the stubhead label to make it clear that each column represents a
 ↪  different year.
tab_stubhead(label = "Year") |>
# Styling the stubhead cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_stubhead()
  ) |>
# Styling the title cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_title()
) |>
# Styling the title cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle"
```

```
    ),

    cell_fill(
    color = "#D6D6D6"
    )
  ),
  locations = cells_column_labels()
) |>
# Styling the title cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      style = "italic",
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_row_groups()
) |>
# Styling the body cells to make them prettier.
tab_style(
  style = cell_text(align = "center"),
  locations = cells_body()
) |>
# Setting other general table options for prettier visualisation.
tab_options(
  table.font.size = px(14),
  heading.align = "center",
  row_group.font.weight = "bold",
  heading.subtitle.font.size = px(16),
  heading.title.font.size = px(18),
  column_labels.hidden = FALSE
) |>
# Setting the source note to comply with editorial requirements.
tab_source_note(
  source_note = "Note: Men only. Occupations are classified using 2-digit
   ↪  ISCO-88 (2007-10) or ISCO-08 (2011-22).\nSource: Own calculations,
   ↪  EU-SILC 2007-22 (N = 1.838.684); weighted sample."
) |>
# Styling the source notes cells to make them prettier.
```

```
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_source_notes()
  )

# Saving the `gt` table as a `.png` file.
gtsave(tablea27b_gt, "Visuals/Tables/TableA27b.png")
```

Table A2.7b. Yearly Trends in Occupational Concentration (1-digit ISCO)

among Men, by Migrant Background

Year	*2007*	*2008*	*2009*	*2010*	*2011*	*2012*	*2015*	*2018*	*2019*	*2020*	*2021*	*2022*
Non-EU Migrant												
Managers	2.27%	1.94%	2.32%	0.98%	1.28%	1.42%	1.43%	1.12%	1.41%	1.34%	2.45%	2.99%
Professionals	2.55%	3.53%	4.01%	4.02%	3.68%	4.56%	6.41%	8.84%	8.78%	10.86%	15.68%	15.72%
Technicians	2.78%	2.60%	4.63%	4.31%	4.41%	5.12%	5.61%	5.74%	6.36%	6.60%	10.20%	9.30%
Clerical workers	2.24%	2.38%	2.46%	2.29%	2.00%	4.01%	2.73%	3.14%	3.24%	3.36%	5.50%	6.26%
Service workers	8.10%	9.42%	10.26%	11.83%	12.54%	14.45%	17.13%	17.70%	15.19%	14.21%	14.00%	15.40%
Skilled workers	4.37%	3.93%	3.02%	4.06%	3.38%	4.51%	4.06%	3.13%	3.62%	2.59%	2.10%	2.06%
Craft and related trades workers	36.05%	34.63%	33.00%	30.80%	30.93%	33.30%	26.31%	24.22%	23.64%	23.10%	17.81%	19.76%
Operators	11.96%	13.12%	13.54%	17.57%	15.08%	10.04%	11.14%	10.98%	12.13%	11.87%	13.42%	12.43%
Elementary workers	29.67%	28.45%	26.76%	24.14%	26.70%	22.59%	25.18%	25.13%	25.62%	26.06%	18.85%	16.08%
EU Migrant												
Managers	5.71%	4.46%	4.21%	6.25%	6.06%	5.64%	5.33%	5.37%	5.25%	4.94%	4.02%	4.87%
Professionals	13.99%	11.55%	10.49%	9.42%	11.05%	12.47%	12.39%	16.17%	18.81%	17.96%	21.47%	20.51%
Technicians	9.46%	7.81%	8.94%	9.45%	6.76%	8.27%	8.46%	8.98%	11.79%	13.77%	13.43%	11.87%
Clerical workers	4.80%	3.47%	5.93%	5.19%	3.92%	3.61%	3.51%	4.94%	4.67%	3.86%	5.71%	5.75%
Service workers	7.91%	9.83%	7.40%	9.07%	9.59%	11.45%	13.29%	10.98%	10.95%	10.95%	9.96%	9.01%
Skilled workers	2.88%	2.01%	3.09%	3.50%	3.21%	3.89%	3.15%	1.46%	1.67%	2.83%	2.02%	1.48%
Craft and related trades workers	30.37%	30.57%	30.96%	29.27%	31.87%	26.08%	27.08%	25.67%	20.55%	20.47%	20.41%	22.22%
Operators	13.64%	17.43%	13.56%	13.00%	12.43%	12.15%	11.93%	11.36%	12.47%	13.22%	13.87%	14.25%
Elementary workers	11.23%	12.86%	15.43%	14.85%	15.12%	16.44%	14.86%	15.05%	13.84%	12.01%	9.11%	10.04%
Native												
Managers	5.71%	5.85%	6.19%	6.17%	5.52%	6.03%	5.45%	5.76%	5.25%	4.54%	4.70%	5.87%
Professionals	11.74%	12.19%	12.15%	12.37%	12.37%	13.50%	15.26%	16.62%	16.64%	18.74%	20.54%	20.44%
Technicians	13.76%	14.29%	14.52%	14.47%	14.23%	14.87%	14.42%	14.86%	14.94%	14.53%	15.15%	14.78%
Clerical workers	7.04%	7.21%	6.97%	7.08%	6.66%	6.62%	6.35%	6.12%	6.34%	6.55%	6.61%	7.09%
Service workers	8.27%	8.14%	8.16%	8.45%	9.32%	10.59%	11.00%	11.07%	11.18%	11.10%	10.50%	10.24%
Skilled workers	2.21%	2.07%	2.14%	2.13%	2.00%	1.97%	2.17%	2.06%	2.07%	1.78%	1.67%	1.61%
Craft and related trades workers	25.10%	24.91%	24.44%	23.76%	22.74%	22.39%	21.53%	19.82%	19.62%	19.24%	18.69%	18.26%
Operators	15.78%	15.31%	15.58%	15.74%	16.37%	14.30%	14.33%	14.45%	15.18%	14.65%	14.32%	13.72%
Elementary workers	10.39%	10.03%	9.84%	9.81%	10.79%	9.74%	9.49%	9.24%	8.78%	8.86%	7.82%	8.00%

Note: Men only. Occupations are classified using 2-digit ISCO-88 (2007-10) or ISCO-08 (2011-22). Source: Own calculations, EU-SILC 2007-22 (N = 1.838.684); weighted sample.

3.7 Table 2.1 / Table 2.2

Tables 2.1 and **2.2** derive from the raw format tibble that synthesises the information contained in **Table A2.7**.

```
table21 <- tablea27 |>
  # Isolating all years delimiting the Great Recession and the COVID-19
   ↪ pandemic, using the filter() `tidyverse` function.
  filter(year %in% c("2007", "2009", "2012", "2019", "2020", "2022")) |>
  # Pivoting the table to a wider format, assigning a separate column of
   ↪ percentages to each year.
  pivot_wider(names_from = year, values_from = perc) |>
  # Calculating all the short-term and long-term percentage changes.
  mutate(
    change_07_09 = (`2009` - `2007`), # Great Recession (short-term).
    change_07_12 = (`2012` - `2007`), # Great Recession (long-term).
    change_19_20 = (`2020` - `2019`), # COVID-19 pandemic (short-term).
    change_19_22 = (`2022` - `2019`) # COVID-19 pandemic (long-term).
  ) |>
  # Pivoting the table to a wider format, assigning a single column to all
   ↪ percentage changes.
  pivot_longer(cols = c(change_07_09, change_07_12,
                        change_19_20, change_19_22),
               names_to = "period",
               values_to = "change") |>
  # Dropping all superfluous variables, using the select() `tidyverse`
   ↪ function.
  select(sexmig, sex, mig, isco, period, change) |>
  # Recoding all variable labels to enhance their readability, using
   ↪ fct_recode() in a mutate() call.
  mutate(
    period = fct_recode(as.factor(period),
                        "Great Recession (2007-09, short-term)" =
                          "change_07_09",
                        "Great Recession (2007-12, long-term)" =
                          "change_07_12",
                        "COVID-19 (2019-20, short-term)" =
                          "change_19_20",
                        "COVID-19 (2019-22, long-term)" =
                          "change_19_22"),

    isco = factor(isco, levels = c("1", "2", "3", "4", "5", "6", "7", "8",
     ↪ "9"),
```

```
                  labels = c("Managers", "Professionals", "Technicians",
                             "Clerical workers", "Service workers",
                             "Skilled workers",
                             "Craft and related trades workers",
                             "Operators", "Elementary workers")))
```

Table 2.1 is appropriately styled and saved as a `.png` file using the `gt` package.

```
# Table 2.1. Changes in Occupational Concentration (1-digit ISCO) during the
 ↪  Great Recession, by Sex and Migrant Background.

table21_gt <- table21 |>
  # Keeping Great Recession time periods only, using the filter() `tidyverse`
   ↪  function.
  filter(period %in% c("Great Recession (2007-09, short-term)",
                       "Great Recession (2007-12, long-term)")) |>
  # Dropping all superfluous variables, using the select() `tidyverse`
   ↪  function.
  select(-sex, -mig) |>
  # Recoding all percentage changes as strings, so that `gt` does not mess
   ↪  with their visualisation.
  mutate(change = if_else(change > 0,
                          sprintf("+%.2f%%", change),
                          sprintf("%.2f%%", change))) |>
  # Pivoting the table to a wider format, assigning a separate column of
   ↪  percentage changes to each sex and migrant background category.
  pivot_wider(names_from = sexmig, values_from = change) |>
  # Building the `gt` table, setting row names as occupational categories,
   ↪  and grouping rows by time period.
  gt(rowname_col = "isco", groupname_col = "period") |>
  # Assigning an appropriate title and subtitle to the `gt` table.
  tab_header(
    title = md("**Table 2.1. Changes in Occupational Concentration (1-digit
     ↪  ISCO)**"),
    subtitle = md("*by Sex and Migrant Background, during the Great
     ↪  Recession*")
  ) |>
  # Imposing the FiveThirtyEight (538) theme on the `gt` table.
  gt_theme_538() |>
  # Fixing the stubhead label to make it clear that each row represents a
   ↪  different occupational category.
  tab_stubhead(label = "Occupation\n(1-digit ISCO)") |>
```

```
  # Styling the stubhead cells to make them prettier.
  tab_style(
    style = list(
      cell_text(
        weight = "bold",
        style = "italic",
        align = "center",
        v_align = "middle",
        whitespace = "pre-wrap",
        transform = "capitalize"
      ),

      cell_fill(
        color = "#D6D6D6"
      )
    ),
    locations = cells_stubhead()
    ) |>
  # Styling the title cells to make them prettier.
  tab_style(
    style = list(
      cell_text(
        align = "center"
      ),

      cell_fill(
        color = "#D6D6D6"
      )
    ),
    locations = cells_title()
  ) |>
  # Styling the column labels cells to make them prettier.
  tab_style(
    style = list(
      cell_text(
        weight = "bold",
        style = "italic",
        align = "center",
        v_align = "middle",
        whitespace = "pre-wrap",
        transform = "capitalize"
      ),
      cell_fill(
```

```
      color = "#D6D6D6"
    )
  ),
  locations = cells_column_labels()
) |>
# Styling the row grouping cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      style = "italic",
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_row_groups()
) |>
# Styling the body cells to make them prettier.
tab_style(
  style = cell_text(align = "center"),
  locations = cells_body()
) |>
# Setting other general table options for prettier visualisation.
tab_options(
  table.font.size = px(14),
  heading.align = "center",
  row_group.font.weight = "bold",
  heading.subtitle.font.size = px(16),
  heading.title.font.size = px(18)
) |>
# Setting the source note to comply with editorial requirements.
tab_source_note(
  source_note = "Note: Occupations are classified using 2-digit ISCO-88
   ↪  (2007-10) or ISCO-08 (2011-22).\nSource: Own calculations, EU-SILC
   ↪  2007-22 (N = 1.838.684); weighted sample."
) |>
# Styling the source notes cells to make them prettier.
tab_style(
  style = cell_text(align = "center"),
  locations = cells_source_notes()
)
```

```
# Saving the `gt` table as a `.png` file.
gtsave(table21_gt, "Visuals/Tables/Table21.png", vwidth = 2200)
```

Table 2.1. Changes in Occupational Concentration (1-digit ISCO)

by Sex and Migrant Background, during the Great Recession

Occupation (1-Digit ISCO)	*Non-EU Migrant Women*	*EU Migrant Women*	*Native Women*	*Non-EU Migrant Men*	*EU Migrant Men*	*Native Men*
			Great Recession (2007-09, short-term)			
Managers	+0.77%	+0.35%	+0.33%	+0.05%	-1.50%	+0.48%
Professionals	-0.27%	-3.82%	+0.96%	+1.46%	-3.50%	+0.41%
Technicians	+2.60%	-1.14%	+0.98%	+1.85%	-0.52%	+0.76%
Clerical workers	+0.88%	-3.19%	-0.97%	+0.22%	+1.13%	-0.07%
Service workers	-3.69%	+3.67%	-0.04%	+2.16%	-0.51%	-0.11%
Skilled workers	-0.23%	-0.05%	+0.02%	-1.35%	+0.21%	-0.07%
Craft and related trades workers	-0.48%	+1.72%	-1.04%	-3.05%	+0.59%	-0.66%
Operators	+0.93%	+1.36%	+0.29%	+1.58%	-0.08%	-0.20%
Elementary workers	-0.50%	+1.10%	-0.52%	-2.91%	+4.20%	-0.55%
			Great Recession (2007-12, long-term)			
Managers	+0.20%	+0.02%	+0.51%	-0.85%	-0.07%	+0.32%
Professionals	+3.56%	-0.66%	+4.55%	+2.01%	-1.52%	+1.76%
Technicians	+2.69%	-1.90%	-1.13%	+2.34%	-1.19%	+1.11%
Clerical workers	-1.06%	-4.47%	-2.90%	+1.77%	-1.19%	-0.42%
Service workers	+2.31%	+4.11%	+1.52%	+6.35%	+3.54%	+2.32%
Skilled workers	+0.86%	+1.27%	+0.11%	+0.14%	+1.01%	-0.24%
Craft and related trades workers	-0.03%	+0.73%	-1.16%	-2.75%	-4.29%	-2.71%
Operators	-1.51%	+1.10%	-0.22%	-1.92%	-1.49%	-1.48%
Elementary workers	-7.02%	-0.23%	-1.28%	-7.08%	+5.21%	-0.65%

Note: Occupations are classified using 2-digit ISCO-88 (2007-10) or ISCO-08 (2011-22). Source: Own calculations, EU-SILC 2007-22 (N = 1.838.684); weighted sample.

Table 2.2 is appropriately styled and saved as a `.png` file using the `gt` package.

```
# Table 2.2. Changes in Occupational Concentration (1-digit ISCO) during the
 ↪  COVID-19 Pandemic, by Sex and Migrant Background.

table22_gt <- table21 |>
  # Keeping COVID-19 time periods only, using the filter() `tidyverse`
   ↪  function.
```

```
filter(period %in% c("COVID-19 (2019-20, short-term)", "COVID-19 (2019-22,
 ↪  long-term)")) |>
# Dropping all superfluous variables, using the select() `tidyverse`
 ↪  function.
select(-sex, -mig) |>
# Recoding all percentage changes as strings, so that `gt` does not mess
 ↪  with their visualisation.
mutate(change = if_else(change > 0,
                        sprintf("+%.2f%%", change),
                        sprintf("%.2f%%", change))) |>
# Pivoting the table to a wider format, assigning a separate column of
 ↪  percentage changes to each sex and migrant background category.
pivot_wider(names_from = sexmig, values_from = change) |>
# Building the `gt` table, setting row names as occupational categories,
 ↪  and grouping rows by time period.
gt(rowname_col = "isco", groupname_col = "period") |>
# Assigning an appropriate title and subtitle to the `gt` table.
tab_header(
  title = md("**Table 2.2. Changes in Occupational Concentration (1-digit
   ↪  ISCO)**"),
  subtitle = md("*by Sex and Migrant Background, during the COVID-19
   ↪  pandemic*")
) |>
# Imposing the FiveThirtyEight (538) theme on the `gt` table.
gt_theme_538() |>
# Fixing the stubhead label to make it clear that each row represents a
 ↪  different occupational category.
tab_stubhead(label = "Occupation\n(1-digit ISCO)") |>
# Styling the stubhead cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle",
      whitespace = "pre-wrap",
      transform = "capitalize"
    ),

    cell_fill(
      color = "#D6D6D6"
```

```
    )
  ),
  locations = cells_stubhead()
  ) |>
# Styling the title cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_title()
) |>
# Styling the column labels cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle",
      whitespace = "pre-wrap",
      transform = "capitalize"
    ),
    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_column_labels()
) |>
# Styling the row grouping cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      style = "italic",
      align = "center"
    ),

    cell_fill(
```

```
        color = "#D6D6D6"
      )
    ),
    locations = cells_row_groups()
  ) |>
  # Styling the body cells to make them prettier.
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_body()
  ) |>
  # Setting other general table options for prettier visualisation.
  tab_options(
    table.font.size = px(14),
    heading.align = "center",
    row_group.font.weight = "bold",
    heading.subtitle.font.size = px(16),
    heading.title.font.size = px(18)
  ) |>
  # Setting the source note to comply with editorial requirements.
  tab_source_note(
    source_note = "Note: Occupations are classified using 2-digit ISCO-88
    ↪  (2007-10) or ISCO-08 (2011-22).\nSource: Own calculations, EU-SILC
    ↪  2007-22 (N = 1.838.684); weighted sample."
  ) |>
  # Styling the source notes cells to make them prettier.
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_source_notes()
  )

# Saving the `gt` table as a `.png` file.
gtsave(table22_gt, "Visuals/Tables/Table22.png", vwidth = 2200)
```

Table 2.2. Changes in Occupational Concentration (1-digit ISCO)

by Sex and Migrant Background, during the COVID-19 pandemic

Occupation (1-Digit ISCO)	*Non-EU Migrant Women*	*EU Migrant Women*	*Native Women*	*Non-EU Migrant Men*	*EU Migrant Men*	*Native Men*
			COVID-19 (2019-20, short-term)			
Managers	-0.42%	-0.38%	-0.34%	-0.07%	-0.31%	-0.71%
Professionals	+0.58%	+1.84%	+1.69%	+2.08%	-0.85%	+2.10%
Technicians	+0.45%	-0.32%	+0.06%	+0.24%	+1.98%	-0.41%
Clerical workers	+0.24%	+0.22%	+0.22%	+0.12%	-0.81%	+0.21%
Service workers	-0.07%	+0.90%	-1.06%	-0.98%	0.00%	-0.08%
Skilled workers	+0.14%	+0.60%	+0.01%	-1.03%	+1.16%	-0.29%
Craft and related trades workers	-0.34%	+0.39%	-0.07%	-0.54%	-0.08%	-0.38%
Operators	-0.67%	-3.25%	-0.25%	-0.26%	+0.75%	-0.53%
Elementary workers	+0.09%	+0.01%	-0.28%	+0.44%	-1.83%	+0.08%
			COVID-19 (2019-22, long-term)			
Managers	+1.06%	-0.49%	+0.76%	+1.58%	-0.38%	+0.62%
Professionals	+5.34%	+7.12%	+3.96%	+6.94%	+1.70%	+3.80%
Technicians	+1.87%	+1.86%	+1.41%	+2.94%	+0.08%	-0.16%
Clerical workers	+1.93%	+4.21%	+0.77%	+3.02%	+1.08%	+0.75%
Service workers	-2.37%	-1.16%	-3.43%	+0.21%	-1.94%	-0.94%
Skilled workers	-0.64%	+0.74%	+0.07%	-1.56%	-0.19%	-0.46%
Craft and related trades workers	-1.07%	+0.03%	-0.31%	-3.88%	+1.67%	-1.36%
Operators	-0.53%	-3.13%	-0.54%	+0.30%	+1.78%	-1.46%
Elementary workers	-5.58%	-9.17%	-2.69%	-9.54%	-3.80%	-0.78%

Note: Occupations are classified using 2-digit ISCO-88 (2007-10) or ISCO-08 (2011-22). Source: Own calculations, EU-SILC 2007-22 (N = 1.838.684); weighted sample.

3.8 Table A2.8 / Table A2.9 / Figure 2.4

Table A2.9 and **Figure 2.4** derive from the raw format tibble that synthesises the information contained in **Table A2.8**.

```
tablea28 <- d_emp |>
  # Isolating all years delimiting the COVID-19 pandemic, using the filter()
   ↪  `tidyverse` function.
  filter(year %in% c("2018", "2019", "2020", "2021", "2022")) |>
  # Recoding all variable labels to enhance their readability, using
   ↪  fct_recode() in a mutate() call.
  select(year, sexmig, occrit, RB050) |>
```

```
  # Recoding all sex and migrant background categories to enhance their
   ↪ readability, using fct_recode() in a mutate() call.
  mutate(
    sexmig = fct_recode(sexmig,
                        "Non-EU Migrant Women" = "NonEUMigWo",
                        "EU Migrant Women" = "EUMigWo",
                        "Native Women" = "NatWo",
                        "Non-EU Migrant Men" = "NonEUMigMe",
                        "EU Migrant Men" = "EUMigMe",
                        "Native Men" = "NatMe"
    )) |>
  # Applying design weights with the as_survey() function from the `srvyr`
   ↪ package.
  as_survey(weights = RB050) |>
  # Grouping by year, sex and migrant background, and employment in an
   ↪ essential occupation.
  group_by(year, sexmig, occrit) |>
  # Counting the weighted number of respondents within each group.
  summarise(weighted_n = survey_total()) |>
  # Computing percentages, rounded to two decimal places.
  mutate(perc = round(weighted_n / sum(weighted_n) * 100, 2)) |>
  # Resetting the grouping.
  ungroup() |>
  # Removing the weighted counts and their associated standard errors.
  select(-weighted_n, -weighted_n_se)
```

Table A2.8 is appropriately styled and saved as a `.png` file using the `gt` package.

```
# Table A2.8. Yearly Trends in Rates of Employment in Essential Occupations,
 ↪ by Sex and Migrant Background.

tablea28_gt <- tablea28 |>
  # Recoding all percentages and years as strings in a mutate() call, so that
   ↪ `gt` does not mess with their visualisation.
  mutate(perc = sprintf("%.2f%%", perc),
         year = as.character(year)) |>
  # Pivoting the table to a wider format, assigning a separate column of
   ↪ percentages to each year.
  pivot_wider(names_from = year, values_from = perc) |>
  # Building the `gt` table, setting row names as essential or non-essential
   ↪ job categories, and grouping rows by sex and migrant background.
  gt(rowname_col = "occrit", groupname_col = "sexmig") |>
```

```
# Assigning an appropriate title and subtitle to the `gt` table.
tab_header(
  title = md("**Table A2.8. Yearly Trends in Rates of Employment in
   ↪  Essential Occupations**"),
  subtitle = md("*by Sex and Migrant Background*")
) |>
# Setting column labels as strings, so that `gt` does not mess with their
 ↪  visualisation.
cols_label(
  `2018` = "2018", `2019` = "2019",
  `2020` = "2020", `2021` = "2021", `2022` = "2022"
) |>
# Imposing the FiveThirtyEight (538) theme on the `gt` table.
gt_theme_538() |>
# Fixing the stubhead label to make it clear that each column represents a
 ↪  different year.
tab_stubhead(label = "Year") |>
# Styling the stubhead cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_stubhead()
  ) |>
# Styling the title cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
```

```
  ),
  locations = cells_title()
) |>
# Styling the column labels cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center"
    ),

    cell_fill(
    color = "#D6D6D6"
    )
  ),
  locations = cells_column_labels()
) |>
# Styling the row grouping cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      style = "italic",
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_row_groups()
) |>
# Styling the body cells to make them prettier.
tab_style(
  style = cell_text(align = "center"),
  locations = cells_body()
) |>
# Setting other general table options for prettier visualisation.
tab_options(
  table.font.size = px(14),
  heading.align = "center",
  row_group.font.weight = "bold",
  heading.subtitle.font.size = px(16),
```

```
    heading.title.font.size = px(18),
    column_labels.hidden = FALSE
  ) |>
  # Setting the source note to comply with editorial requirements.
  tab_source_note(
    source_note = "Note: Only employed respondents.\nSource: Own
    ↪  calculations, EU-SILC 2007-22 (N = 1.296.594); weighted sample."
  ) |>
  # Styling the source notes cells to make them prettier.
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_source_notes()
  )

# Saving the `gt` table as a `.png` file.
gtsave(tablea28_gt, "Visuals/Tables/TableA28.png")
```

Table A2.8. Yearly Trends in Rates of Employment in Essential Occupations

by Sex and Migrant Background

Year	2018	2019	2020	2021	2022
Non-EU Migrant Women					
Non-essential occupation	48.82%	46.79%	44.08%	48.63%	48.62%
Essential occupation	51.18%	53.21%	55.92%	51.37%	51.38%
EU Migrant Women					
Non-essential occupation	55.49%	55.10%	52.37%	55.64%	58.25%
Essential occupation	44.51%	44.90%	47.63%	44.36%	41.75%
Native Women					
Non-essential occupation	60.35%	60.39%	60.54%	61.63%	61.59%
Essential occupation	39.65%	39.61%	39.46%	38.37%	38.41%
Non-EU Migrant Men					
Non-essential occupation	60.71%	59.56%	55.02%	55.87%	59.10%
Essential occupation	39.29%	40.44%	44.98%	44.13%	40.90%
EU Migrant Men					
Non-essential occupation	65.15%	60.11%	58.79%	61.67%	65.85%
Essential occupation	34.85%	39.89%	41.21%	38.33%	34.15%
Native Men					
Non-essential occupation	60.88%	61.15%	60.23%	60.58%	61.21%
Essential occupation	39.12%	38.85%	39.77%	39.42%	38.79%

Note: Only employed respondents. Source: Own calculations, EU-SILC 2007-22 (N = 1.296.594); weighted sample.

Table A2.9 is appropriately styled and saved as a `.png` file using the `gt` package.

```
# Table A2.9. Changes in Rates of Employment in Essential Occupations, by Sex
 ↪  and Migrant Background.

tablea29_gt <- tablea28 |>
  # Isolating all years delimiting the COVID-19 pandemic, using the filter()
   ↪  `tidyverse` function.
  filter(year %in% c("2019", "2020", "2022")) |>
  # Pivoting the table to a wider format, assigning a separate column of
   ↪  percentages to each year.
  pivot_wider(names_from = year, values_from = perc) |>
  # Calculating all the short-term and long-term percentage changes.
  mutate(
    change_19_20 = (`2020` - `2019`), # COVID-19 pandemic (short-term).
    change_19_22 = (`2022` - `2019`) # COVID-19 pandemic (long-term).
  ) |>
  # Pivoting the table to a wider format, assigning a single column to all
   ↪  percentage changes.
  pivot_longer(cols = c(change_19_20, change_19_22),
               names_to = "period",
               values_to = "change") |>
  # Dropping all superfluous variables, using the select() `tidyverse`
   ↪  function.
  select(sexmig, occrit, period, change) |>
  # Recoding all variable labels to enhance their readability, using
   ↪  fct_recode() in a mutate() call.
  mutate(
    period = fct_recode(as.factor(period),
                        "COVID-19\n(2019-20, short-term)" =
                          "change_19_20",
                        "COVID-19\n(2019-22, long-term)" =
                          "change_19_22")) |>
  # Recoding all percentage changes as strings, so that `gt` does not mess
   ↪  with their visualisation.
  mutate(change = if_else(change > 0,
                          sprintf("+%.2f%%", change),
                          sprintf("%.2f%%", change))) |>
  # Keeping only respondents in essential occupations using the filter()
   ↪  `tidyverse` function.
  filter(occrit %in% "Essential occupation") |>
  # Dropping the now superfluous essential occupation identifier, using the
   ↪  select() `tidyverse` function.
  select(-occrit) |>
```

```
# Pivoting the table to a wider format, assigning a separate column of
 ↪  percentage changes to each time period category.
pivot_wider(names_from = period, values_from = change) |>
# Building the `gt` table, setting row names as sex and migrant background
 ↪  category.
gt(rowname_col = "sexmig") |>
# Assigning an appropriate title and subtitle to the `gt` table.
tab_header(
  title = md("**Table A2.9. Changes in Rates of Employment in Essential
   ↪  Occupations**"),
  subtitle = md("*by Sex and Migrant Background*")
) |>
# Imposing the FiveThirtyEight (538) theme on the `gt` table.
gt_theme_538() |>
# Fixing the stubhead label to make it clear that each column represents a
 ↪  different time period.
tab_stubhead(label = "Time Period") |>
# Styling the stubhead cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_stubhead()
  ) |>
# Styling the title cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
```

```
    )
  ),
  locations = cells_title()
) |>
# Styling the column labels cells to make them prettier.
tab_style(
  style = list(
    cell_text(
      weight = "bold",
      style = "italic",
      align = "center",
      v_align = "middle",
      whitespace = "pre-wrap",
      transform = "capitalize"
    ),
    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_column_labels()
) |>
tab_style(
  # Styling the row grouping cells to make them prettier.
  style = list(
    cell_text(
      style = "italic",
      align = "center"
    ),

    cell_fill(
      color = "#D6D6D6"
    )
  ),
  locations = cells_row_groups()
) |>
# Styling the body cells to make them prettier.
tab_style(
  style = cell_text(align = "center"),
  locations = cells_body()
) |>
# Setting other general table options for prettier visualisation.
tab_options(
  table.font.size = px(14),
```

```
    heading.align = "center",
    row_group.font.weight = "bold",
    heading.subtitle.font.size = px(16),
    heading.title.font.size = px(18)
  ) |>
  # Setting the source note to comply with editorial requirements.
  tab_source_note(
    source_note = "Note: Only employed respondents.\nSource: Own
     ↪  calculations, EU-SILC 2007-22 (N = 1.296.594); weighted sample."
  ) |>
  # Styling the source notes cells to make them prettier.
  tab_style(
    style = cell_text(align = "center"),
    locations = cells_source_notes()
  )

# Saving the `gt` table as a `.png` file.
gtsave(tablea29_gt, "Visuals/Tables/TableA29.png")
```

Table A2.9. Changes in Rates of Employment in Essential Occupations

by Sex and Migrant Background

Time Period	*COVID-19 (2019-20, Short-Term)*	*COVID-19 (2019-22, Long-Term)*
Non-EU Migrant Women	+2.71%	-1.83%
EU Migrant Women	+2.73%	-3.15%
Native Women	-0.15%	-1.20%
Non-EU Migrant Men	+4.54%	+0.46%
EU Migrant Men	+1.32%	-5.74%
Native Men	+0.92%	-0.06%

Note: Only employed respondents. Source: Own calculations, EU-SILC 2007-22 (N = 1.296.594); weighted sample.

Figure 2.4 is appropriately styled and saved as a `.svg` file using the `ggplot2` package.

```r
# Figure 2.4. Yearly Trends in Employment in Essential Occupations by Sex and
 ↪  Migrant Background.

figure24 <- tablea28 |>
  # Recoding all years as numeric in a mutate() call, so that `ggplot2` can
   ↪  generate a line plot.
  mutate(year = as.numeric(year)) |>
  # Keeping only respondents in essential occupations using the filter()
   ↪  `tidyverse` function.
  filter(occrit %in% c("Essential occupation")) |>
  # Generating a lineplot of essential employment rates by year, sex, and
   ↪  migrant background.
  ggplot(aes(x = year, y = perc, linetype = sexmig, color = sexmig)) +
  # Setting an appropriate line width to facilitate the figure's
   ↪  interpretability.
  geom_line(linewidth = 0.7) +
  # Manually annotating rectangles of differing grey shades to highlight the
   ↪  COVID-19 pandemic.
  annotate("rect", xmin = 10, xmax = 11, ymin = -Inf, ymax = Inf,
           alpha = 0.3, fill = "#A3A3A3") +
  annotate("rect", xmin = 11, xmax = 12, ymin = -Inf, ymax = Inf,
           alpha = 0.3, fill = "#C2C2C2") +
  # Assigning appropriate labels to all rectangles to highlight the COVID-19
   ↪  pandemic.
  geom_richtext(data = tibble(
    year = c(10.5, 11.5),
    perc = c(58.5, 58.5),
    label = c("COVID-19<br>(short-term)", "COVID-19<br>(long-term)")
    ),
    aes(x = year, y = perc, label = label),
    inherit.aes = FALSE, size = 3.5, color = "black", fontface = "bold") +
  # Setting a continuous year-by-year scale on the x-axis.
  scale_x_continuous(breaks = c(8, 9, 10, 11, 12),
                     labels = c("2018", "2019", "2020", "2021", "2022")) +
  # Setting a continuous percentage scale on the y-axis.
  scale_y_continuous(breaks = seq(30, 60, 1), limits = c(33, 59)) +
  # Setting combinations of grey shades and line types to highlight sex and
   ↪  migrant background categories.
  scale_linetype_manual(values = c("solid", "dotted", "twodash",
                                   "solid", "dotted", "twodash")) +
  scale_color_manual(values = c("#000000", "#000000", "#000000",
                                "#999999", "#999999", "#999999")) +
```

```r
  # Styling axes and legends to make the figure look prettier.
  guides(color = guide_legend(byrow = TRUE), linetype = guide_legend(byrow =
   ↪  TRUE)) +
  labs(
    title = "",
    subtitle = "",
    x = "",
    y = "% of Respondents",
    color = "",
    linetype = ""
  ) +
  # Styling the overall theme to make the figure look prettier.
  theme_bw(base_size = 14) +
  theme(
    plot.title.position = "plot",
    plot.title = element_text(hjust = 0.5, face = "bold", size = 18),
    plot.subtitle = element_text(hjust = 0.5, face = "italic", size = 14),
    legend.position = "bottom",
    strip.text = element_text(face = "bold")
  )

# Saving the `ggplot2` figure as a `.svg` file.
ggsave(filename = "Visuals/Figures/Figure24.svg",
       plot = figure24, width = 10, height = 8, dpi = 600)
```

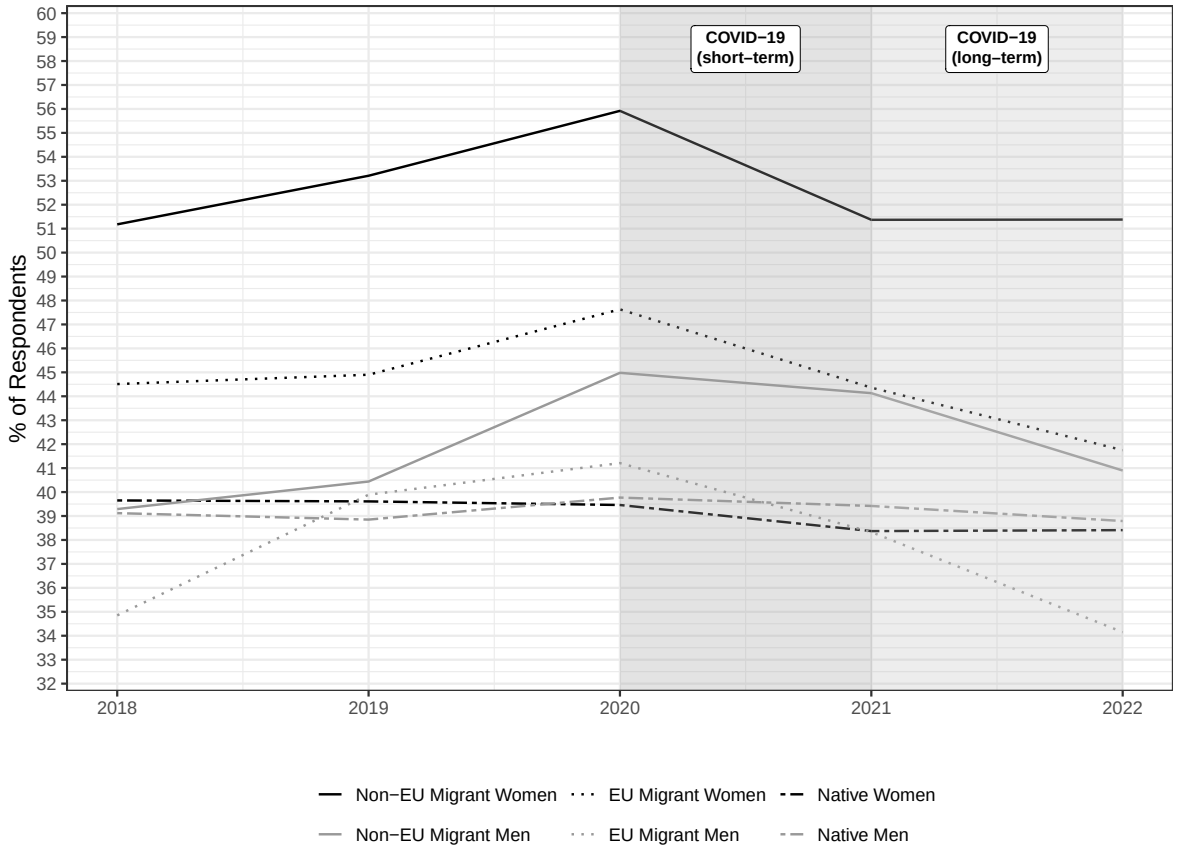

Figure 2.4. *Yearly Trends in Temporary Employment Rates by Sex and Migrant Background.*

4. References

Angulo-Gonzalez, C., Fuentealba-Chaura, R., Mora, R., & Rojas-Mora, J. (2021). *Mutualinf: Computation and Decomposition of the Mutual Information Index*. https://doi.org/10.32614/CRAN.package.mutualinf

Auguie, B. (2010). *gridExtra: Miscellaneous Functions for "Grid" Graphics*. https://doi.org/10.32614/CRAN.package.gridExtra

Bengtsson, H. (2015). *Future: Unified Parallel and Distributed Processing in R for Everyone*. https://doi.org/10.32614/CRAN.package.future

Eurostat. (2024). *EU Statistics on Income and Living Conditions microdata, 2024 release, data covering years 2004-2023*. Eurostat. https://doi.org/10.2907/EUSILC2004-2023

Freedman Ellis, G., & Schneider, B. (2016). *Srvyr: 'Dplyr'-Like Syntax for Summary Statistics of Survey Data*. https://doi.org/10.32614/CRAN.package.srvyr

Guinea-Martin, D., Mora, R., & Ruiz-Castillo, J. (2015). The joint effect of ethnicity and gender on occupational segregation. An approach based on the Mutual Information Index. *Social Science Research*, *49*, 167–178. https://doi.org/10.1016/j.ssresearch.2014.08.007

Iannone, R., Cheng, J., Schloerke, B., Hughes, E., Lauer, A., Seo, J., Brevoort, K., & Roy, O. (2025). *Gt: Easily Create Presentation-Ready Display Tables*. https://gt.rstudio.com

Mock, T. (2022). *gtExtras: Extending 'gt' for Beautiful HTML Tables*. https://doi.org/10.32614/CRAN.package.gtExtras

R Core Team. (2021). *R: A language and environment for statistical computing*. https://www.R-project.org/

Wickham, H. (2016). *ggplot2*. Springer International Publishing. https://doi.org/10.1007/978-3-319-24277-4

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T., Miller, E., Bache, S., Müller, K., Ooms, J., Robinson, D., Seidel, D., Spinu, V., … Yutani, H. (2019). Welcome to the Tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., Miller, E., & Smith, D. (2015). *Haven: Import and Export 'SPSS', 'Stata' and 'SAS' Files*. https://doi.org/10.32614/CRAN.package.haven

Wilke, C. O., & Wiernik, B. M. (2020). *Ggtext: Improved Text Rendering Support for 'ggplot2'*. https://doi.org/10.32614/CRAN.package.ggtext