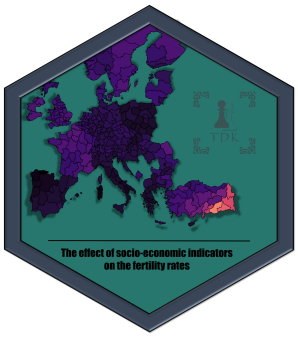


The effect of socio-economic indicators on the fertilty rates

An empirical analysis of fertility rates amoung the regions of Europe

Marcell Granát

February 21, 2021

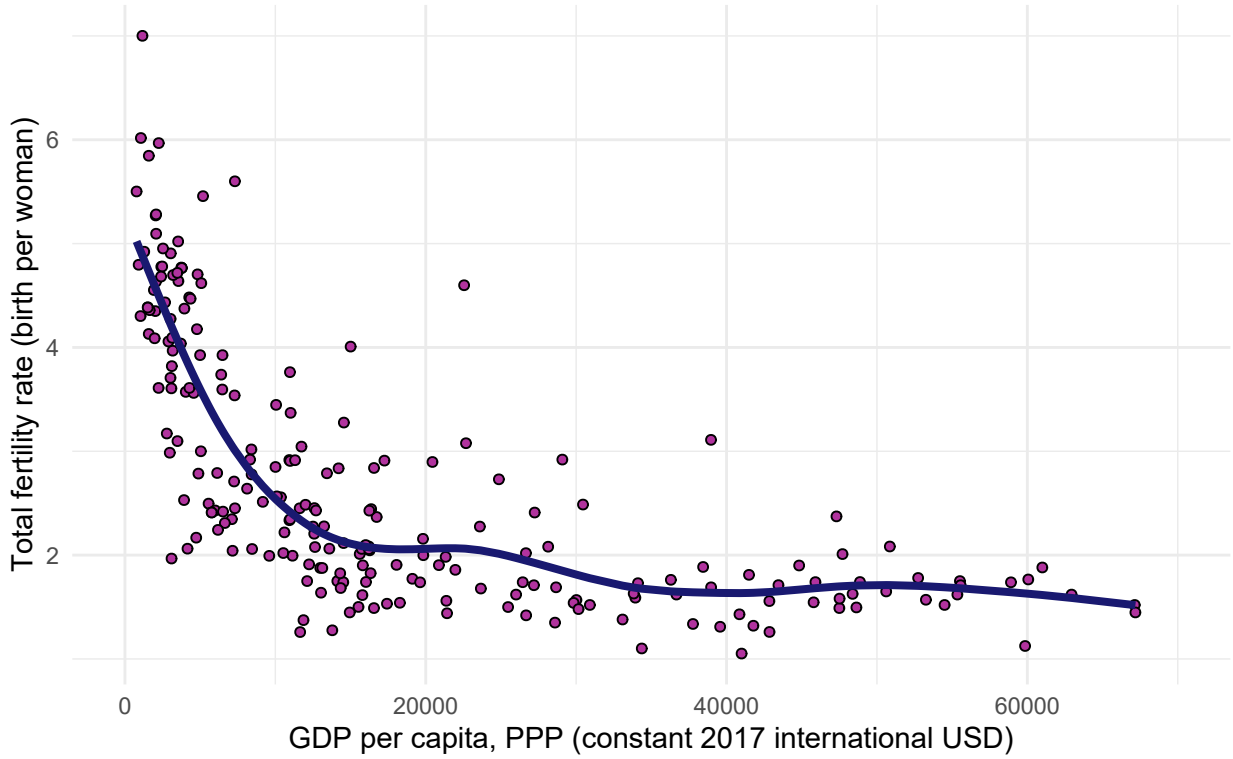
Contents

Abstract

Here is the abstract.

Introduction

Ay alábbi rövid kézirat a brit **Family Expenditure Survey** adataiból kinyert fogyasztási jellemzőket tárgyalja. Az empirikus elemzés egyszerű lineáris regresszió alkalmazásával készül. Az adatok összesen 1519 család jövedelmét és kiadását tartalmazzák, továbbá 6 termékcsoport (alkohol, ruházkodás, nem alkoholtartalmú élelmiszer, fűtés, közlekedés, egyéb) fogyasztáson belüli részarányát. Dolgozatom során én kizárólag az egy gyermekes családokra szűkítem az elemzést[1].



Own editing based on the Figure 5-2. from Kreiszné Hudák (2019).
The trend is drawn via splines.
Source of the data: World Bank.

Figure 1: Seemingly negative effect of gross domestic product on fertility rates based on nation level observations (2017)

Data

Total fertility rates

Human development

Table 1: Indicators of similiarity between the Human Development Indices provided by UNDP and GDL

Indicator	Value
R^2	99.76%
Spearman R^2	99.71%

[1]Feladatkiírás által előírt megkötés.

Indicator	Value
Mean absolute deviation	0.007
Mean absolute percentage deviation	1.20%

A decent standard of living

Table 2: Indicators of similiarity between the income component of the Human Development Indices provided by GDL and the estimation based on regional GDP

Indicator	Value
R^2	92.43%
Spearman R^2	94.54%
Mean absolute deviation	0.0474
Mean absolute percentage deviation	6.00%

Long and healthy life

Table 3: Indicators of similiarity between the health component of the Human Development Indices provided by GDL and the estimation based on regional life expectency

Indicator	Value
R^2	98.24%
Spearman R^2	98.38%
Mean absolute deviation	0.0041
Mean absolute percentage deviation	0.45%

Knowledge

Table 4: Indicators of similiarity between the knowledge component of Human Development Indices provided by UNDP and the calculated principal components using educational attainment level

Indicator	Comp 1	Comp 2	Comp 3	Comp 4	Comp 5	Comp 6	Comp 7	Comp 8	Comp 9
R^2	33.35%	23.35%	10.34%	0.03%	0.61%	0.16%	0.00%	0.05%	0.01%
Spearman R^2	21.15%	23.77%	13.54%	0.03%	0.62%	0.51%	13.29%	0.47%	1.74%

Family benefits

Youth unemployment

Explore data

Model building

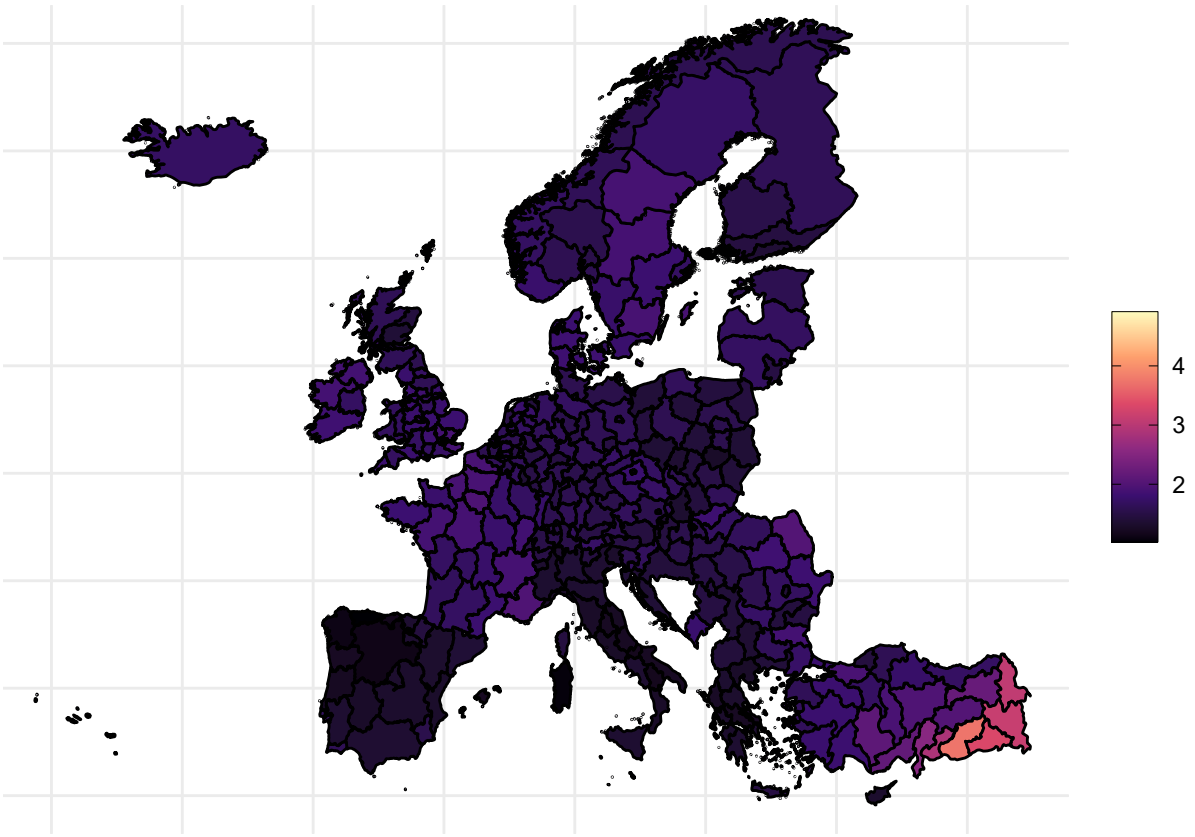


Figure 2: Total fertility rates through Europe in 2017

Figure 3: HDI and its components based on the dataset from Global Data Lab (2017)

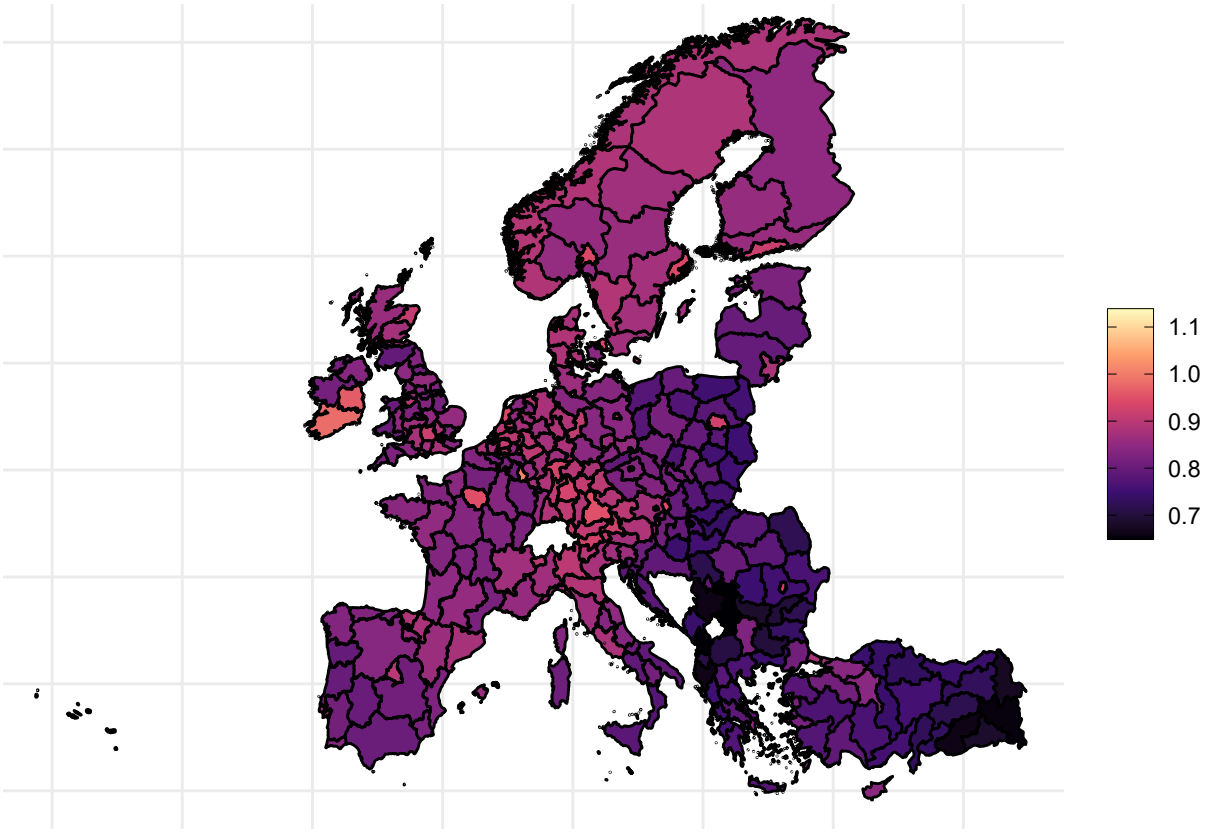


Figure 4: Calculated income index based on regional GDP data

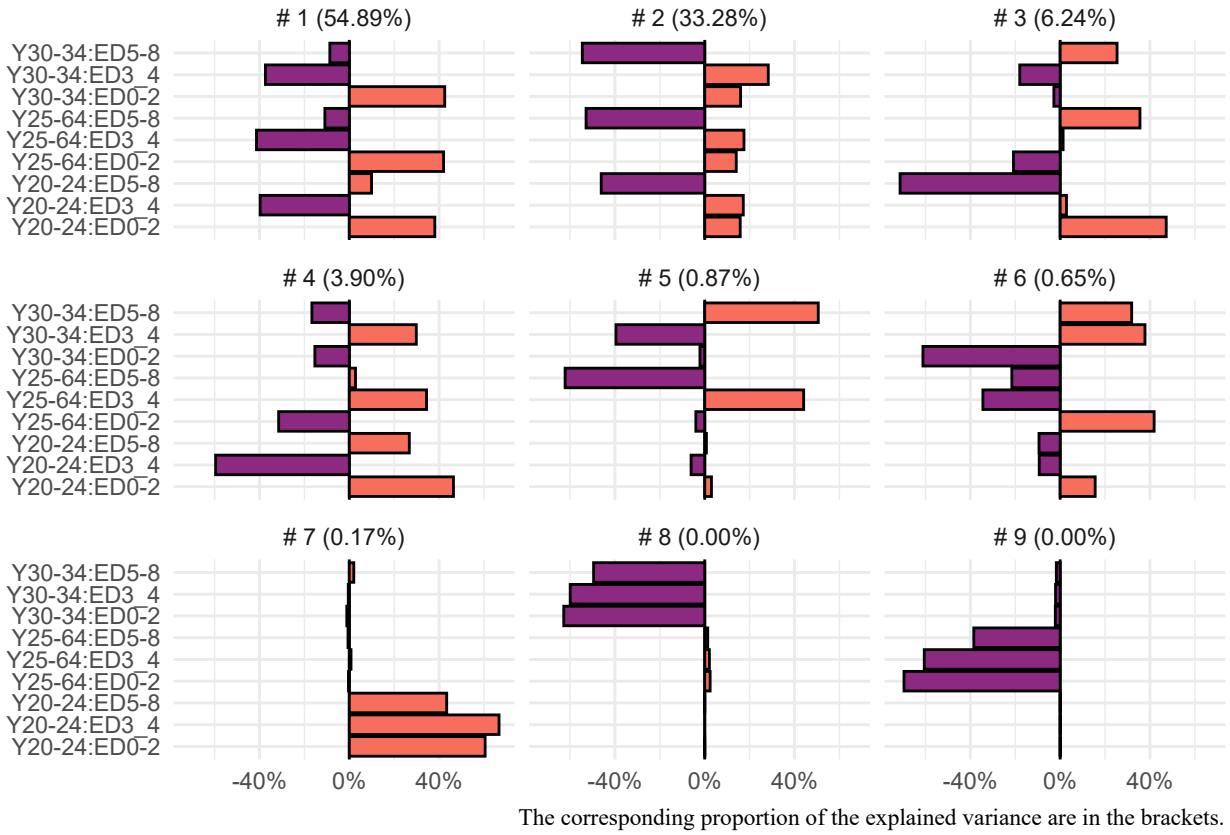


Figure 5: PCAs and the explained variance

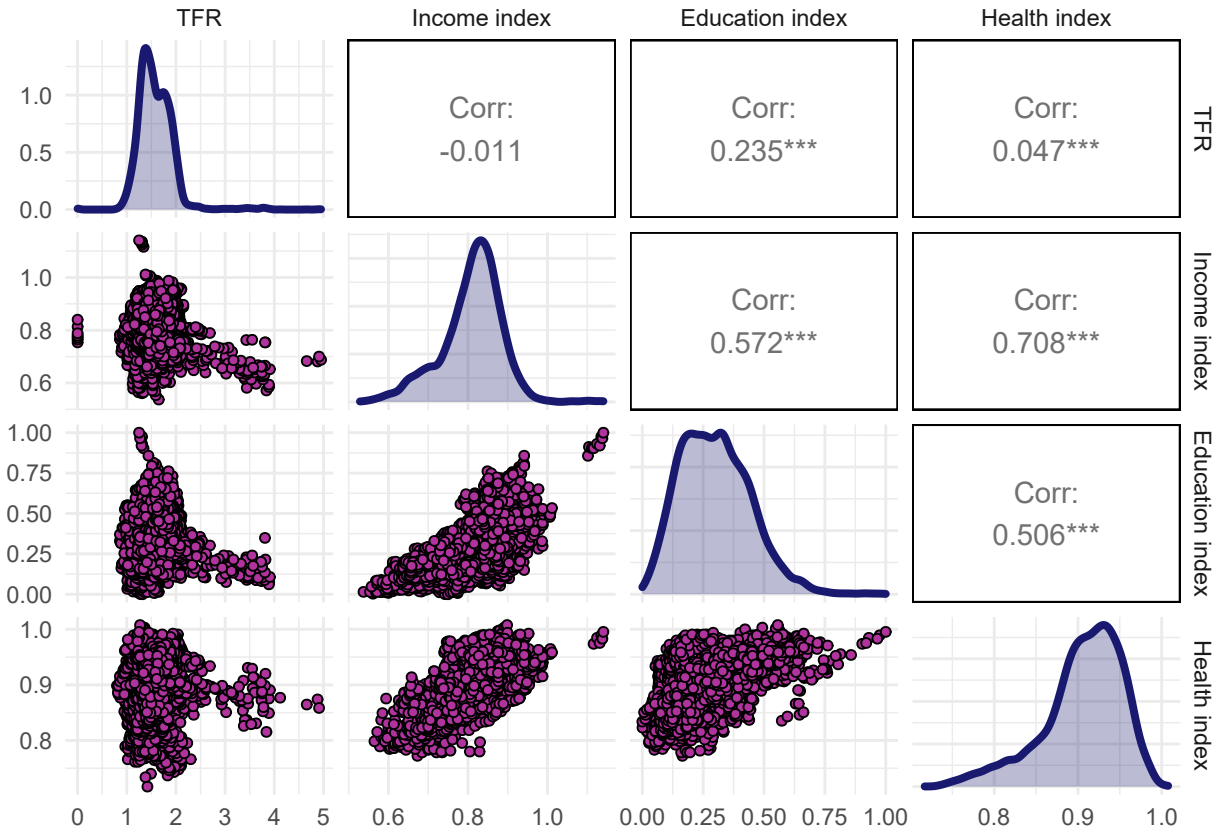


Figure 6: Pairwise correlation among TFR and calculation human development indices

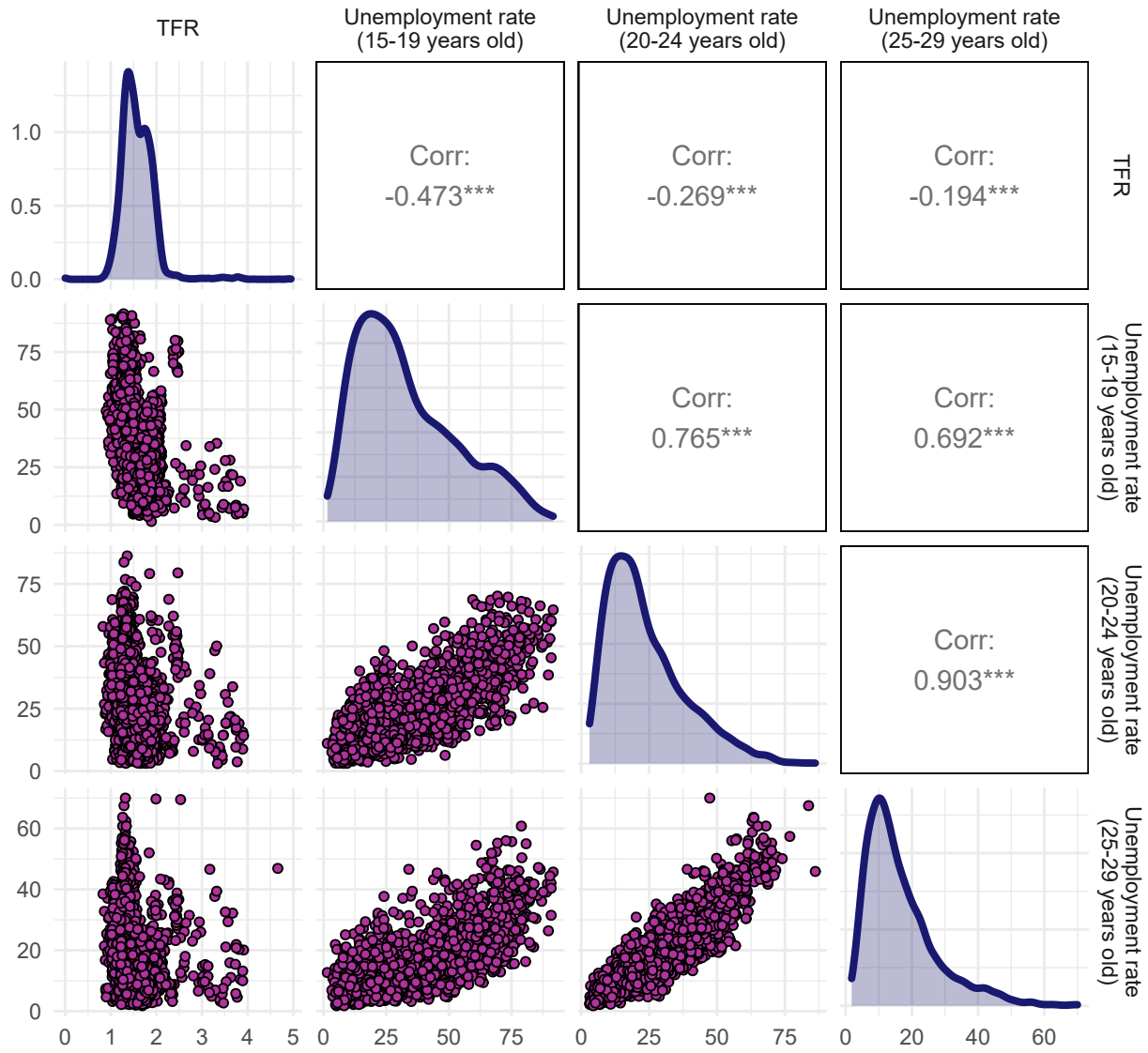


Figure 7: Pairwise correlation among TFR and youth unemployment rates

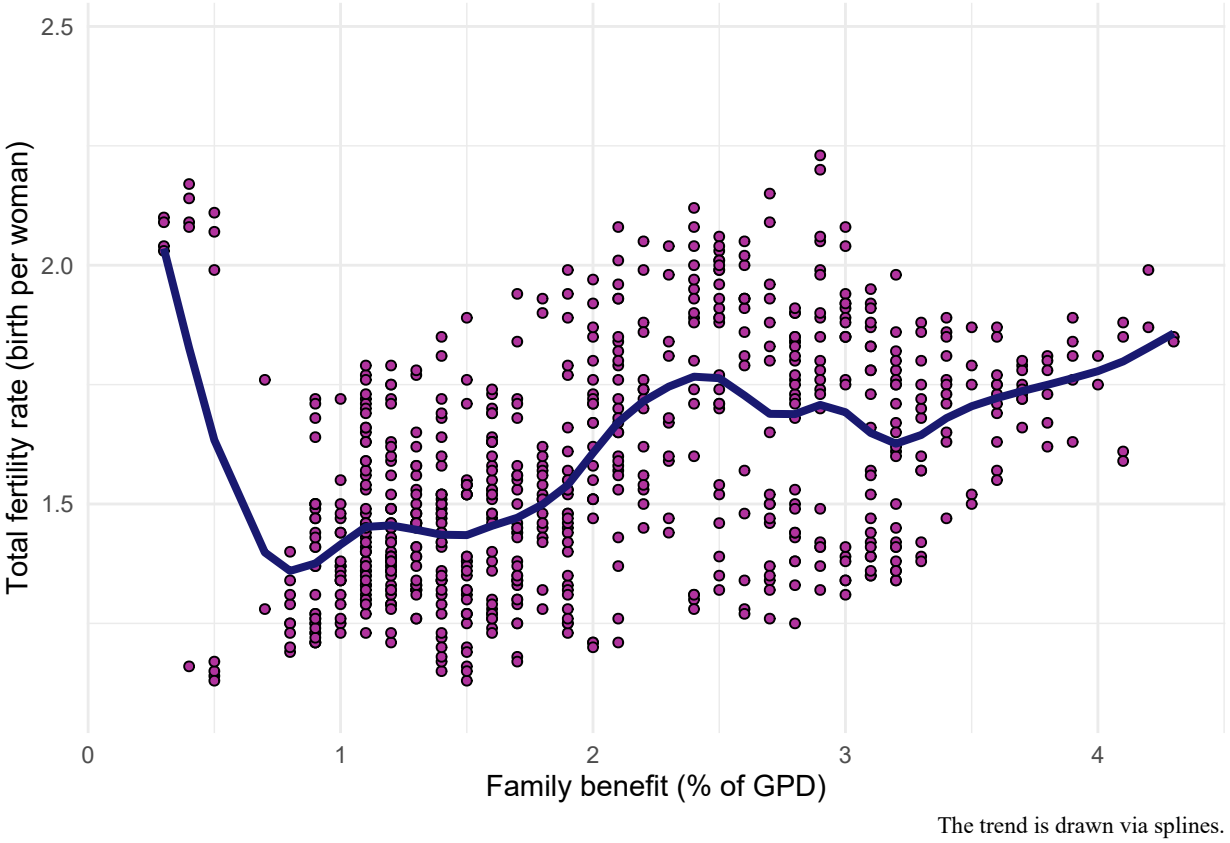


Figure 8: Correlation between TFR and family benefits

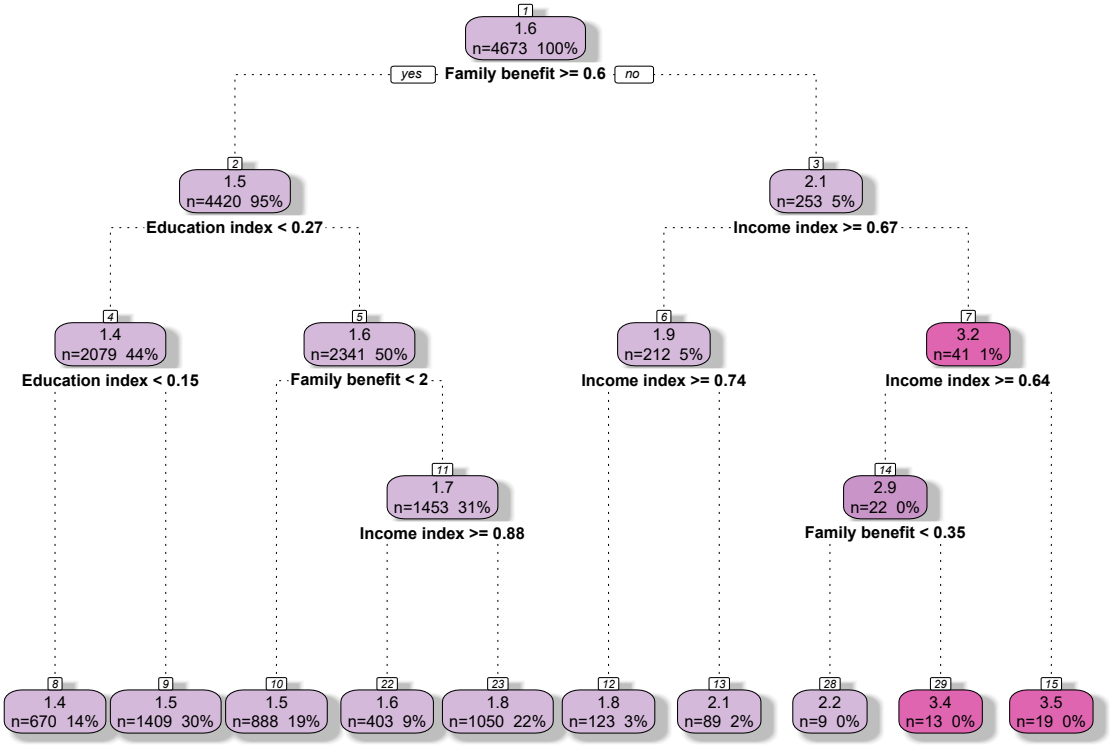


Figure 9: Regression tree explaining the TFR using only the calculated HD indices (cp = 0.01)

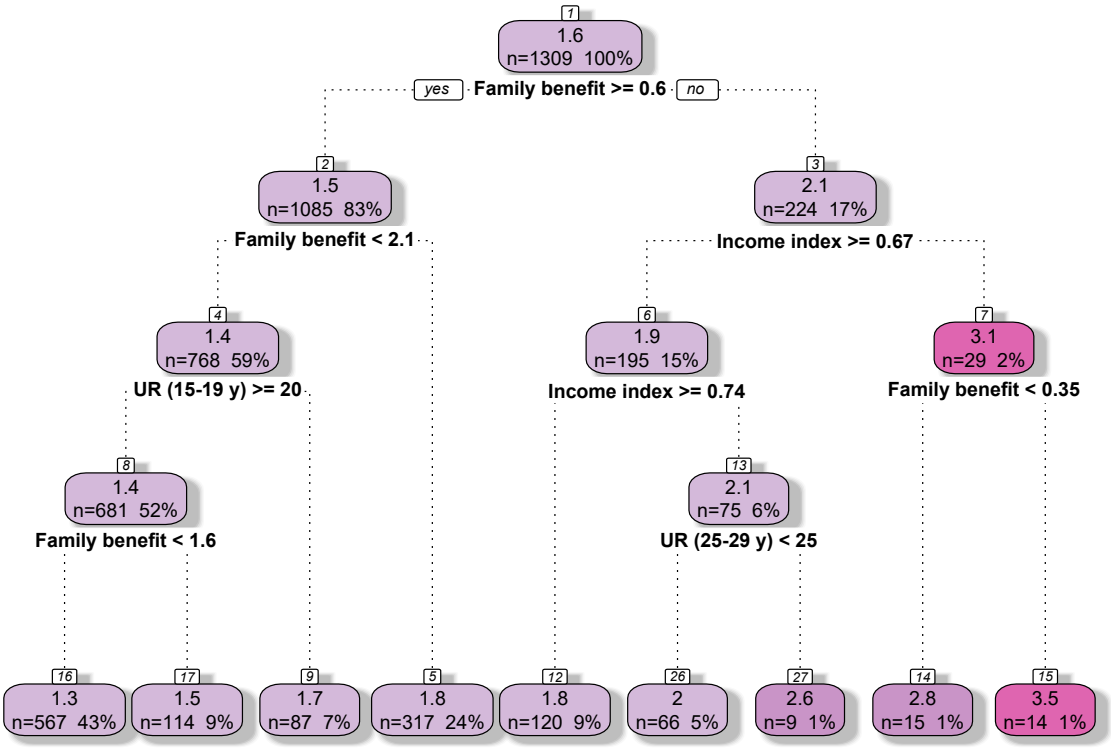


Figure 10: Regression tree explaining the TFR using all the mentioned explanatory variables (cp = 0.01)

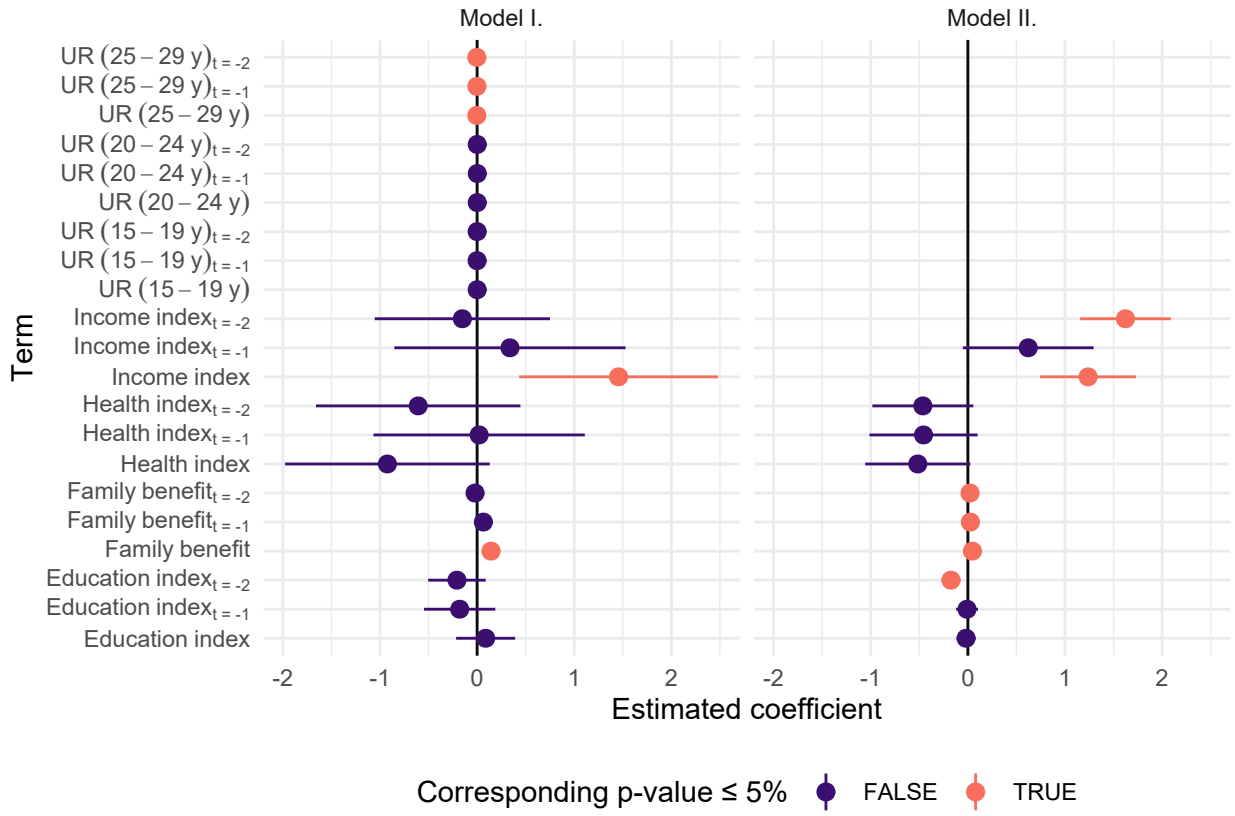


Figure 11: Panel models on the total fertility rates

Table 5: Models

Indicator	Model I.	Model II.
Pooltest	0.00%	0.00%
Phtest	0.00%	0.00%
Adjusted R^2	11.07%	20.06%
Observations	882	4020

Framework I: with unemployment

Table 6: T-tests

Variable	Mean in total sample	Mean in used sample	Number of observations in the total sample	T-statistic	P-value
TFR	1.5799	1.5928	7246	-0.9650	33.48%
Education index	0.2939	0.3131	5375	-3.6811	0.02%
Health index	0.9069	0.9294	6971	-16.3615	0.00%
Income index	0.8125	0.8197	4962	-3.1597	0.16%
Family benefit	2.0073	1.4374	6417	18.7511	0.00%
UR (15-19 y)	34.1416	38.8821	1891	-5.5185	0.00%
UR (20-24 y)	24.2588	24.9203	3088	-1.2496	21.16%
UR (25-29 y)	16.9295	16.7099	2809	0.5654	57.19%

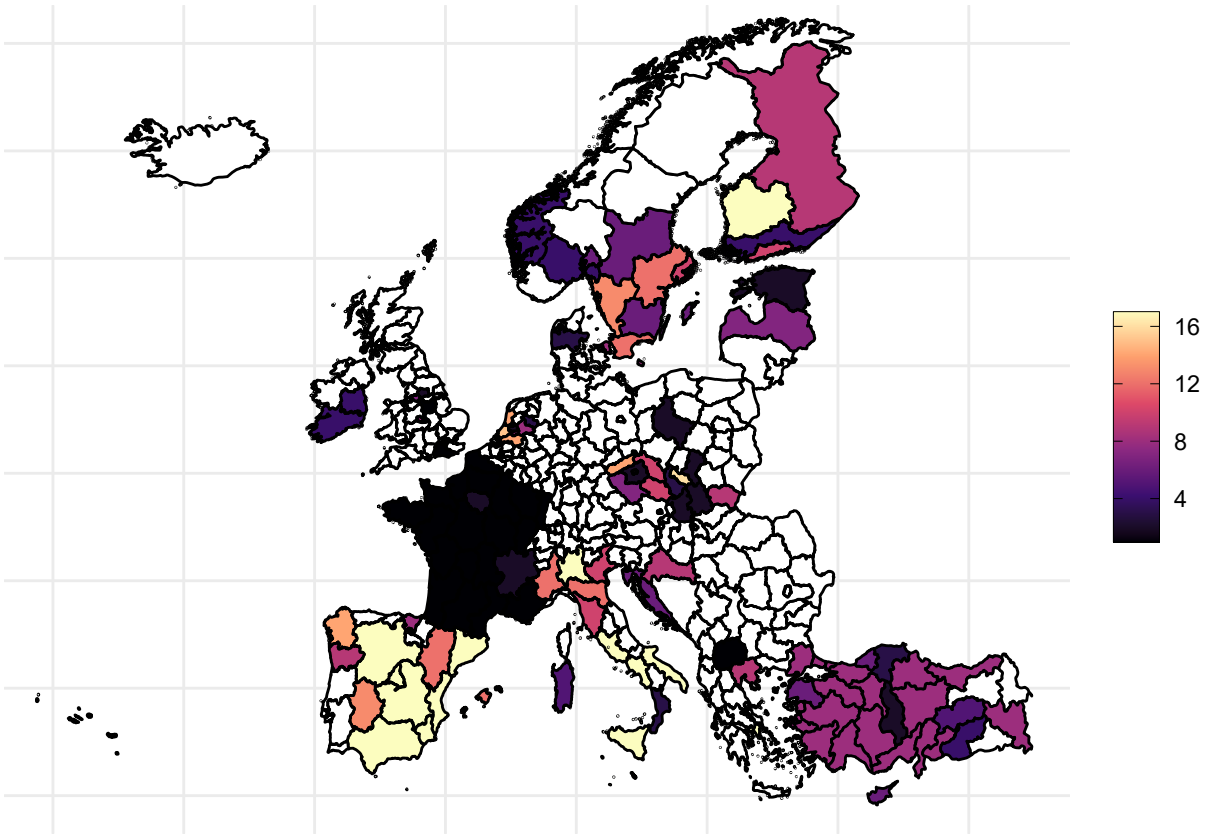
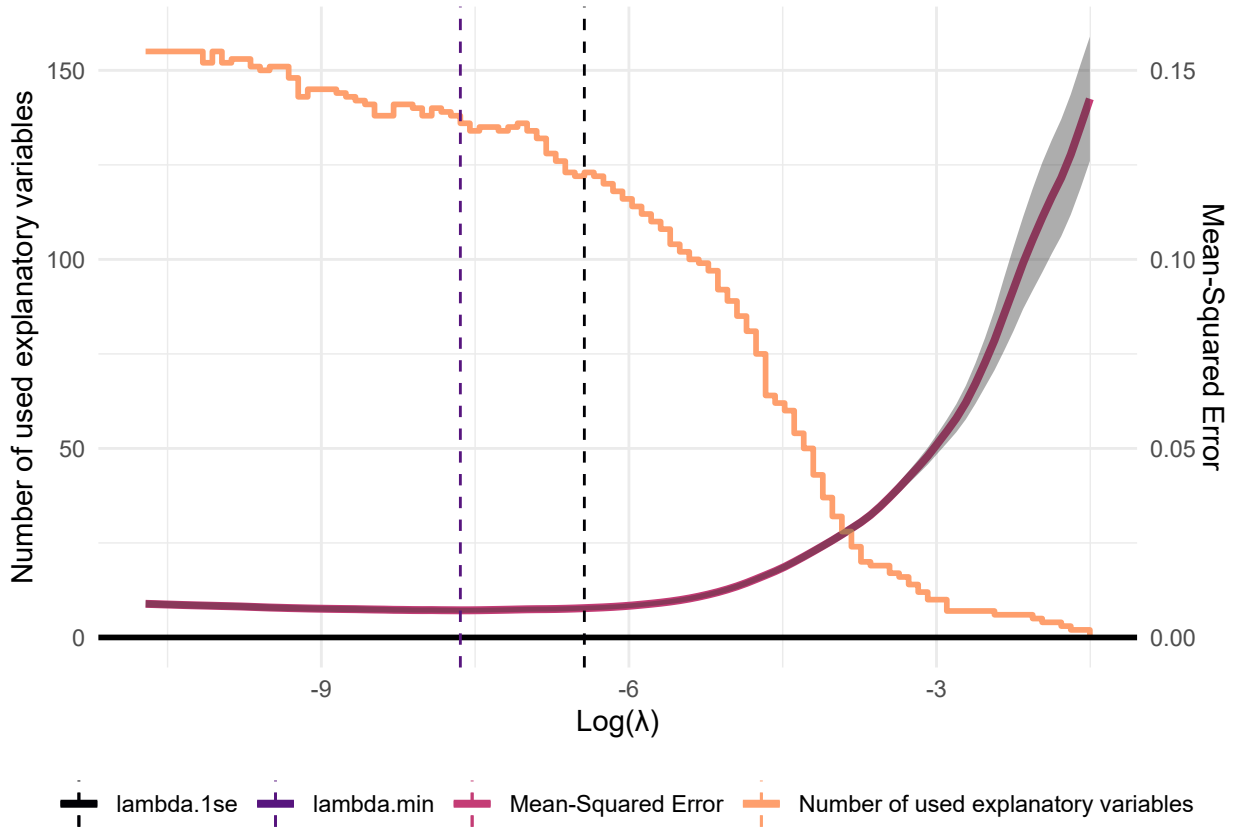


Figure 12: Number of used observations by countries when the model contains youth unemployment



Framework II: without unemployment

Table 7: T-tests

Variable	Mean in total sample	Mean in used sample	Number of observations in total sample	T-statistic	P-value
TFR	1.5799	1.5694	7246	1.6855	9.19%
Education index	0.2939	0.2982	5375	-1.4818	13.84%
Health index	0.9069	0.9202	6971	-16.0011	0.00%
Income index	0.8125	0.8239	4962	-8.2285	0.00%
Family benefit	2.0073	2.0029	6417	0.2365	81.31%

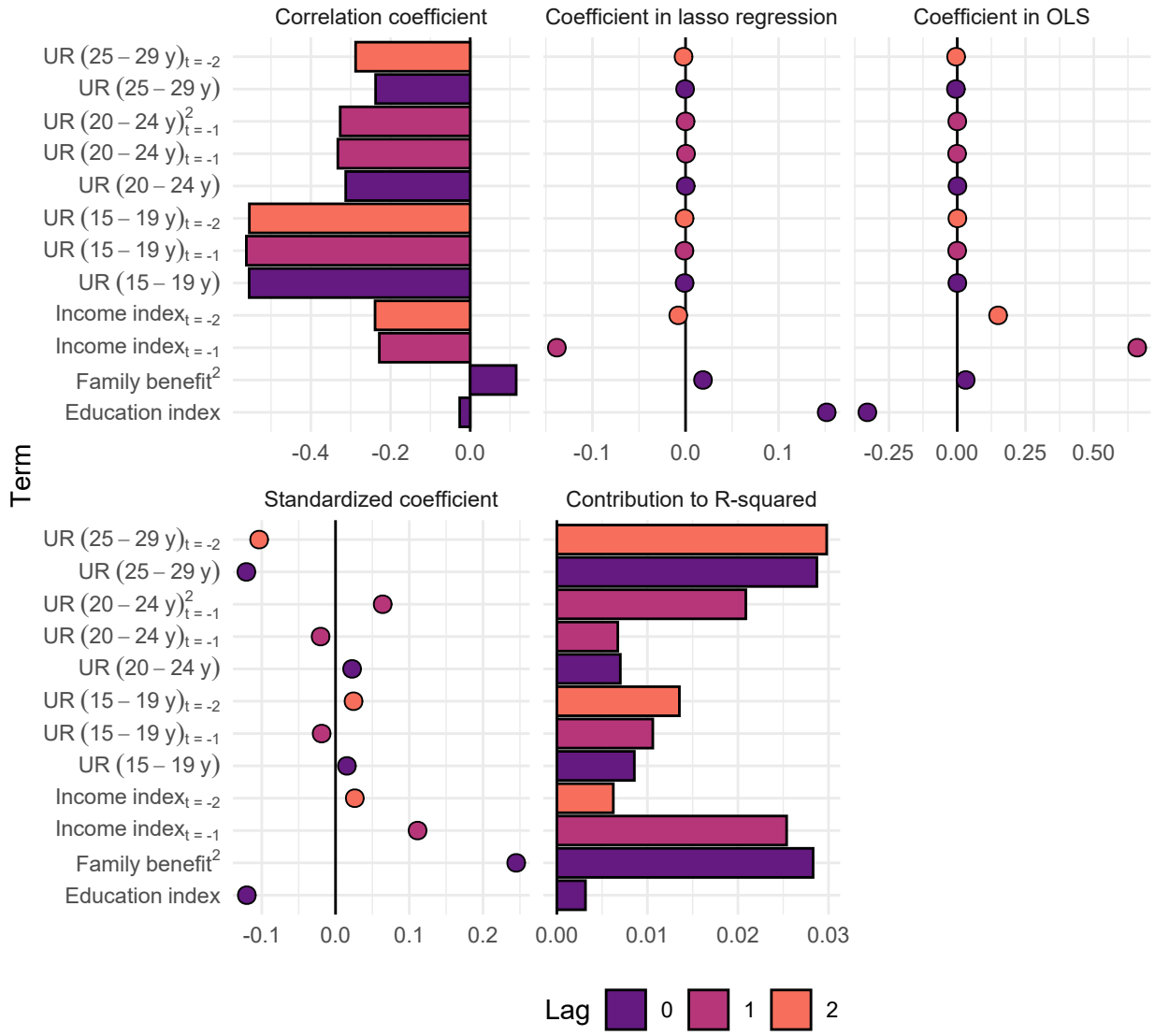


Figure 13: Estimated coefficient of the fixed panel model controlling for youth unemployment indicators

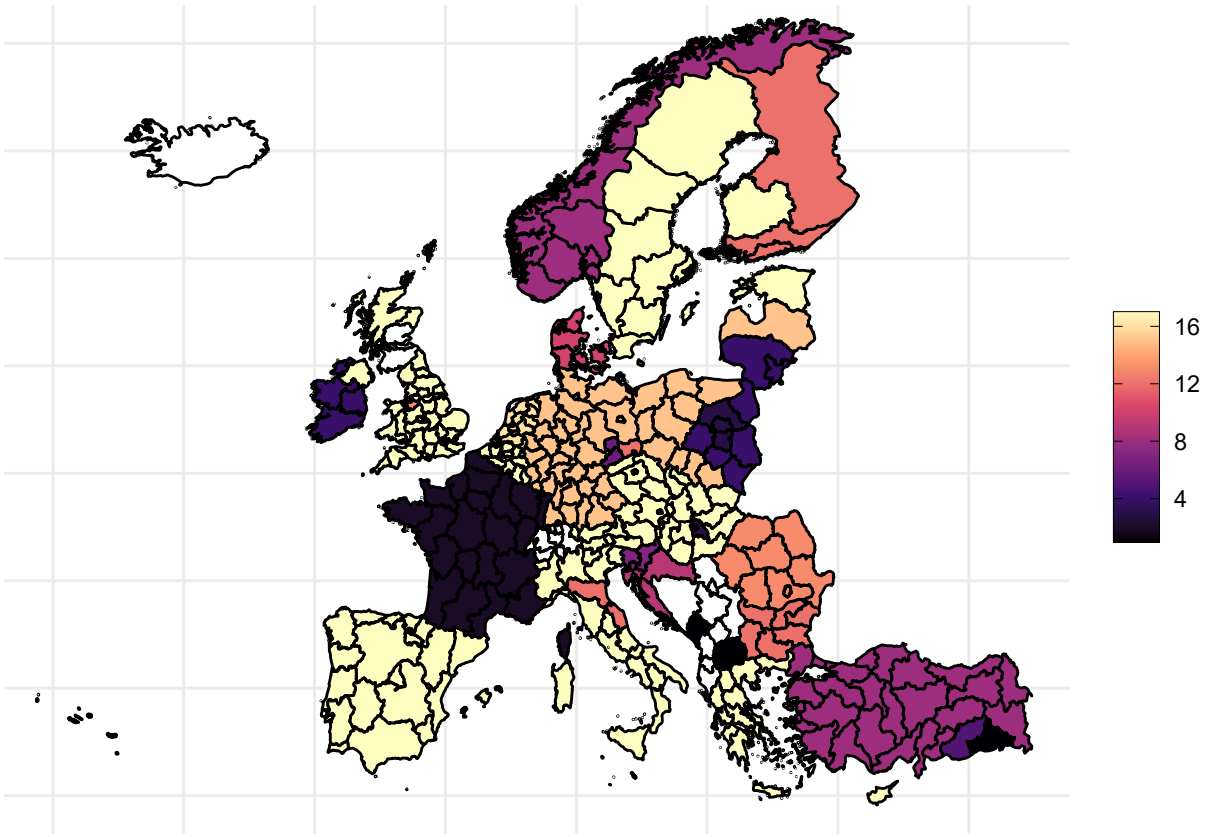


Figure 14: Number of used observations by countries when the model does not contain youth unemployment

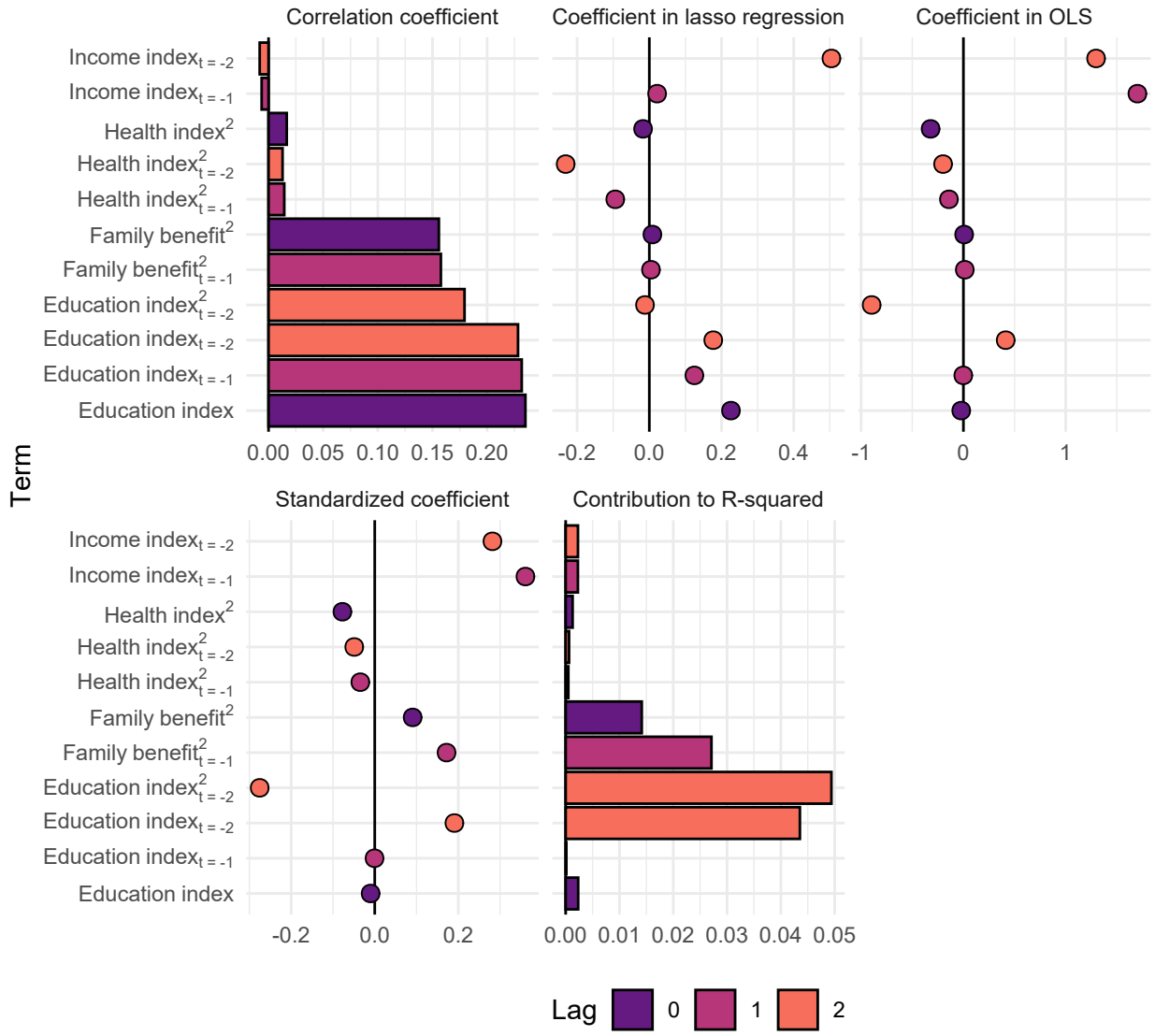


Figure 15: Estimated coefficient of the fixed panel model omitting youth unemployment indicators

References

UNITED NATIONS DEVELOPMENT PROGRAMME (2020), 'Technical notes: Calculating the human development indices'.
URL: *http://hdr.undp.org/en/content/calculating-indices*

Appendix: R codes

```r
# Set up ------------------------------------------------------------------------------

## Packages ==========================================================================

library(tidyverse)
library(patchwork)
library(knitr)
library(broom)
library(eurostat)

## Gg theme ==========================================================================

update_geom_defaults("point", list(fill = "#B1339E",
                                   shape = 21,
                                   color = "black",
                                   size = 1.4))
update_geom_defaults("line",
                     list(color = "midnightblue", size = 1.4))

update_geom_defaults("smooth", list(color = "red4", size = 1.4))

update_geom_defaults("density",
                     list(color = "midnightblue", fill =  "midnightblue", alpha = .3,
                          size = 1.4))

extrafont::loadfonts(device="win")

theme_set(theme_minimal() + theme(
  legend.direction = "vertical",
  # text = element_text(family = "Impact"),
  plot.caption = element_text(family = "serif")
))

# https://data.worldbank.org/indicator/SP.DYN.TFRT.IN

WB_fertility <- read_csv("WB_fertility.csv", skip = 4)

# https://data.worldbank.org/indicator/NY.GDP.PCAP.PP.KD

WB_GDP <- read_csv("WB_GDP.csv", skip = 4)

merge(WB_fertility %>%
        select('Country Name', '2017') %>%
        rename(tfr = '2017'),
      WB_GDP %>%
        select('Country Name', '2017') %>%
        rename(GDP = '2017')) %>%
  ggplot(aes(GDP, tfr)) + geom_point() +
  ggformula::geom_spline() +
  scale_x_continuous(limits = c(0, 7e+4)) +
  labs(y = "Total fertility rate (birth per woman)",
       x = "GDP per capita, PPP (constant 2017 international USD)",
```

```
       caption = "Own editing based on the Figure 5-2. from Kreiszné Hudák (2019).
       The trend is drawn via splines.
       Source of the data: World Bank."
  )

plot_NUTS2 <- function(df, viridis_c = T, ..., all.x = F) {
  p <- df %>%
    {merge(eurostat::get_eurostat_geospatial(nuts_level = 2), ., all.x = all.x)} %>%
    ggplot(aes(fill = values)) +
    geom_sf(color = "black") +
    theme(
      axis.text = element_blank()
    ) +
    xlim(c(-30, 44)) +
    ylim(c(35, 70)) +
    labs(fill = NULL)

  if (viridis_c) {
    p <- p + scale_fill_viridis_c(option = "magma", ...,
                                  guide = guide_colorbar(frame.colour = "black",
                                                         ticks.colour = "black"),
                                  na.value = "white")
  }
  p
}

# Data import ------------------------------------------------------------------

# Fertility: source: Eurostat database ===========================================

f_data <- get_eurostat("demo_r_find2", time_format = "num") %>%
  select(geo, time, var = indic_de, values)

### Map of TFR ##################################################################

f_data %>%
  filter(var == "TOTFERRT" & str_length(geo) == 4 & time == 2017) %>%
  plot_NUTS2()

# Data from Global Data Labor ====================================================

### Sub-national data ###########################################################

# source of csv files: https://globaldatalab.org/

GDL_import <- function(x) {
  get_eurostat_geospatial(nuts_level = 2) %>%
    data.frame() %>%
    tibble %>%
    mutate(
      ISO_Code = countrycode::countrycode(CNTR_CODE, origin = "iso2c", "iso3c"),
      ISO_Code = ifelse(CNTR_CODE == "UK", "GBR", ISO_Code),
      ISO_Code = ifelse(CNTR_CODE == "EL", "GRC", ISO_Code),
```

```
    ) %>%
    select(ISO_Code, NUTS_NAME, geo) %>%
    merge(read_csv(x), by = "ISO_Code") %>%
    mutate(
      z = stringdist::stringsim(NUTS_NAME, Region)
    ) %>%
    arrange(desc(z)) %>%
    filter(!duplicated(Region)) %>%
    filter(!duplicated(NUTS_NAME)) %>%
    filter((z > .5 | Country %in% c("Greece", "Turkey", 'Romania',
                                    'Malta', 'Italy')) & NUTS_NAME != "Dresden") %>%
    select(geo, '1990':'2018') %>%
    pivot_longer(-1, names_to = "time", values_to = "values") %>%
    mutate(time = as.numeric(time))
}

GDL_subnat <- GDL_import("GDL-Sub-national-HDI-data.csv") %>% rename(HDI = values) %>%
  merge(
    GDL_import("GDL-Educational-index--data.csv") %>% rename(education = values)
  ) %>%
  merge(
    GDL_import("GDL-Health-index-data.csv") %>% rename(health = values)
  ) %>%
  merge(
    GDL_import("GDL-Income-index-data.csv") %>% rename(income = values)
  )

### National data ##############################################################################

GDL_nat <- read_csv("GDL-Sub-national-HDI-data.csv") %>%
  filter(Level == "National") %>%
  select(Country, 6:34) %>% pivot_longer(-1, names_to = "time", values_to = "values") %>%
  mutate(time = as.numeric(time)) %>%
  na.omit()

# Data from UNDP ===============================================================================

# source: http://hdr.undp.org/

HDI_UNDP <- read_csv("Human Development Index (HDI).csv",
                     skip = 5) %>%
  select(!starts_with("X"), - 'HDI Rank') %>%
  mutate_at(-1,
            function(x) {as.numeric(ifelse(x == "..", NA, x))}) %>%
  pivot_longer(-1, names_to = "time", values_to = "values") %>%
  mutate(time = as.numeric(time)) %>%
  na.omit()

merge(GDL_nat %>% rename(GDL = values),
      HDI_UNDP %>% rename(UNDP = values)) %>%
  {
    c(
      scales::percent(cor(x = .$GDL, y = .$UNDP)^2, accuracy = .01),
```

```r
      scales::percent(cor(x = .$GDL, y = .$UNDP, method = "spearman")^2, accuracy = .01),
      as.character(format(mean(abs(.$GDL - .$UNDP)), digits = 1)),
      scales::percent(mean(abs(.$GDL - .$UNDP)/.$UNDP), accuracy = .01)
    )
  } %>%
  {tibble(
    Indicator = c("$R^2$", "Spearman $R^2$",
                  "Mean absolute deviation", "Mean absolute percentage deviation"),
    Value = .
  )} %>%
  kable(
    caption = "Indicators of similiarity between the Human Development Indices
    provided by UNDP and GDL"
  )

GDL_subnat %>%
  filter(time == 2017) %>%
  select(-time) %>%
  pivot_longer(-1, names_to = "var", values_to = "values") %>%
  filter(!is.na(var)) %>%
  mutate(
    var = str_to_title(var),
    var = str_replace(var, "Hdi", "HDI"),
    var = factor(var,
                 levels = c("HDI", "Income", "Education", "Health"),
                 ordered = T)
  ) %>%
  plot_NUTS2() + facet_wrap(~ var, ncol = 2) +
  labs(caption = "Source: https://globaldatalab.org")

GDP_index <- get_eurostat("nama_10r_2gdp", time_format = "num") %>%
  filter(unit == "PPS_HAB") %>%  # Purchasing power standard (PPS) per inhabitant
  select(-unit) %>%
  rename(GDP = values) %>% merge(
    get_eurostat("ert_bil_eur_a", time_format = "num") %>% # EUR/USD annual avg exc r
      filter(currency == "USD" & statinfo == "AVG") %>%
      select(time, e = values)
  ) %>%
  {
    GDP <- .$GDP/.$e # mutate to USD
    mutate(.,
           GDPindex = (log(GDP) - log(100))/
             (log(75000) - log(100))
    )
  }

merge(GDL_subnat, GDP_index) %>%
  {
    c(
      scales::percent(cor(x = .$income, y = .$GDPindex)^2, accuracy = .01),
      scales::percent(cor(x = .$income, y = .$GDPindex, method = "spearman")^2,
                      accuracy = .01),
      as.character(format(mean(abs(.$income - .$GDPindex)), digits = 1, nsmall = 4)),
```

```
      scales::percent(mean(abs(.$income - .$GDPindex)/.$GDPindex), accuracy = .01)
    )
  } %>%
  {tibble(
    Indicator = c("$R^2$", "Spearman $R^2$",
                  "Mean absolute deviation", "Mean absolute percentage deviation"),
    Value = .
  )} %>%
  kable(
    caption = "Indicators of similiarity between the income component of the
    Human Development Indices provided by GDL and the estimation based on regional GDP"
  )

GDP_index %>%
  filter(time == 2017) %>%
  select(geo, values = GDPindex) %>%
  plot_NUTS2

health_index <- get_eurostat("demo_r_mlifexp", time_format = "num") %>%
  filter(age == "Y_LT1" & sex == "T") %>%
  select(geo, time, le = values) %>%
  mutate(
    health_index = (le - 20) / (85 - 20)
  )

merge(GDL_subnat, health_index) %>%
  {
    c(
      scales::percent(cor(x = .$health, y = .$health_index)^2, accuracy = .01),
      scales::percent(cor(x = .$health, y = .$health_index, method = "spearman")^2,
                      accuracy = .01),
      as.character(format(mean(abs(.$health - .$health_index)), digits = 1, nsmall = 4)),
      scales::percent(mean(abs(.$health - .$health_index)/.$health_index), accuracy = .01)
    )
  } %>%
  {tibble(
    Indicator = c("$R^2$", "Spearman $R^2$",
                  "Mean absolute deviation", "Mean absolute percentage deviation"),
    Value = .
  )} %>%
  kable(
    caption = "Indicators of similiarity between the health component of the
    Human Development Indices provided by GDL and the estimation based on regional life
    expectency"
  )

edu_wide <- get_eurostat("edat_lfse_04", time_format = "num") %>%
  filter(sex == "T" & !str_detect(isced11, "GEN") &
           !str_detect(isced11, "VOC") & isced11 != "ED3-8"
  ) %>%
  mutate(
    var = str_c(age, ":", isced11)
  ) %>%
```

```r
  select(geo, time, var, values) %>%
  pivot_wider(names_from = var, values_from = values) %>%
  {
    x <- .
    names(x) <- letters[1:length(x)]
    x <- cbind(
      .[, 1:2],
      mice::complete(mice::mice(select(x, -a,-b), printFlag = F))
    )
    names(x) <- names(.)
    x
  }

edu_comps <- edu_wide%>%
  select(-time, -geo) %>%
  na.omit() %>%
  {princomp(scale(.))}

edu_comp_vars <- edu_comps %>%
  summary() %>%
  {.$sdev^2/sum(.$sdev^2)} %>%
  scales::percent(accuracy = .01) %>%
  {str_c("# ", 1:length(.), " (", ., ")")}

edu_comps %>%
  .$loadings %>%
  unclass() %>%
  data.frame() %>%
  rownames_to_column() %>%
  pivot_longer(-1) %>%
  mutate(
    name = as.numeric(str_remove(name, 'Comp.')),
  ) %>%
  arrange(name) %>%
  mutate(
    name = edu_comps %>%
      summary() %>%
      {.$sdev^2/sum(.$sdev^2)} %>%
      scales::percent(accuracy = .01) %>%
      {str_c("# ", 1:length(.), " (", ., ")")} %>%
      .[name]
  ) %>%
  ggplot +
  aes(rowname, value, fill = value < 0) +
  geom_hline(yintercept = 0) +
  geom_col(color = 'black') +
  coord_flip() +
  scale_fill_viridis_d(guide = F, option = "magma", begin = .4,
                       end= .7, direction = -1) +
  scale_y_continuous(labels = scales::percent) +
  facet_wrap(~name, ncol = 3) +
  labs(x = NULL, y = NULL, caption =
         "The corresponding proportion of the explained variance are in the brackets.")
```

```r

edu_comps %>% .$scores %>%
  cbind(edu_wide) %>% merge(GDL_subnat) %>%
  select(3:11, education) %>%
  {
    x <- .$education
    apply(select(., -education), 2, function(y) {
      c(
        scales::percent(cor(x = x, y = y)^2, accuracy = .01),
        scales::percent(cor(x = x, y = y, method = "spearman")^2, accuracy = .01)
      )
    })
  } %>%
  data.frame() %>%
  mutate(Indicator = c("$R^2$", "Spearman $R^2$")) %>%
  rename_all(function(x) str_replace(x, "p.", "p ")) %>%
  select(Indicator, 1:9) %>%
  kable(
    caption = "Indicators of similiarity between the knowledge component of Human
    Development Indices provided by UNDP and the calculated principal components using
    educational attainment level"
  )

edu_index <- edu_comps %>%
  .$scores %>%
  data.frame() %>%
  select(2) %>%
  cbind(edu_wide) %>%
  select(geo, time, edu_index = Comp.2) %>%
  mutate(
    edu_index = -edu_index,
    edu_index = edu_index + abs(min(edu_index)),
    edu_index = edu_index/max(edu_index)
  )

dat <- f_data %>%
  pivot_wider(names_from = var, values_from = values) %>%
  merge(edu_index, all = T) %>%
  merge(health_index, all = T) %>%
  merge(GDP_index, all = T)

FAM_df <- get_eurostat("spr_exp_sum", time_format = "num") %>%
  filter(spdeps == "FAM" & unit == "PC_GDP") %>%
  rename(FAM = values, country = geo) %>%
  select(-(spdeps:unit))

yth_empl_byage <- get_eurostat("yth_empl_110", time_format = "num") %>%
  filter(unit == "PC" & sex == "F") %>%
  filter(age %in% c("Y15-19", "Y20-24", "Y25-29")) %>%
  select(-unit, -sex) %>%
  pivot_wider(names_from = age, values_from = values) %>%
  rename(
    "uY15"  = "Y15-19",
    "uY20"  = "Y20-24",
```

```
    "uY25"  = "Y25-29"
  )

dat <- dat %>%
  mutate(country = str_sub(geo, end = 2)) %>%
  merge(FAM_df, all.x = T, all.y = F) %>%
  merge(yth_empl_byage, all.x = T, all.y = F)

f.clean_names <- function(v, Tosparse = F) {
  v <- str_replace_all(v, "GDPindex", "Income index") %>%
    str_replace_all("health_index", "Health index") %>%
    str_replace_all("edu_index", "Education index") %>%
    str_replace_all("TOTFERRT", "TFR") %>%
    str_replace_all("GDPindex", "Income index") %>%
    str_replace_all("FAM", "Family benefit") %>%
    str_replace_all("uY15", "UR (15-19 y)") %>%
    str_replace_all("uY20", "UR (20-24 y)") %>%
    str_replace_all("uY25", "UR (25-29 y)")
  if(Tosparse) v <- str_replace_all(v, " ", "~")
  v
}

# Explore the data ------------------------------------------------------------

# Pairwise correlations ========================================================

dat %>%
  filter(str_length(geo) == 4) %>%
  select(TOTFERRT, GDPindex, edu_index, health_index) %>%
  {set_names(., f.clean_names(names(.)))} %>%
  GGally::ggpairs()

dat %>%
  filter(str_length(geo) == 4) %>%
  select(TOTFERRT, uY15, uY20, uY25) %>%
  set_names("TFR", "Unemployment rate\n(15-19 years old)",
            "Unemployment rate\n(20-24 years old)",
            "Unemployment rate\n(25-29 years old)") %>%
  GGally::ggpairs()

dat %>%
  filter(str_length(geo) == 2) %>%
  ggplot(aes(FAM, TOTFERRT)) + geom_point() +
  ggformula::geom_spline() +
  labs(y = "Total fertility rate (birth per woman)",
       x = "Family benefit (% of GPD)",
       caption = "The trend is drawn via splines."
  )

# Regression trees =============================================================

m_part <- dat %>%
  filter(str_length(geo) == 4) %>%
```

```r
  select(TOTFERRT, GDPindex, edu_index, health_index, FAM) %>%
  {set_names(., f.clean_names(names(.)))} %>%
  na.omit() %>%
  rpart::rpart(formula = TFR ~ ., cp = .01)

m_part %>% rattle::fancyRpartPlot(palettes = 'PuRd', sub = NULL)

m_part %>% summary()

m_part2 <- dat %>%
  filter(str_length(geo) == 4) %>%
  select(TOTFERRT, GDPindex, edu_index, health_index, uY15, uY20, uY25, FAM) %>%
  {set_names(., f.clean_names(names(.)))} %>%
  na.omit() %>%
  rpart::rpart(formula = TFR ~ ., cp = .01)

m_part2 %>% rattle::fancyRpartPlot(palettes = 'PuRd', sub = NULL)

# Model building ---------------------------------------------------------------

# Transform the data for panel modeling ==========================================

dat_plm <- dat %>%
  select(
    geo, time, TOTFERRT, edu_index, health_index, GDPindex, FAM, uY15, uY20, uY25
  ) %>%
  filter(str_length(geo) == 4 & !is.na(TOTFERRT))

dat_plm <- dat_plm %>%
  select(-TOTFERRT) %>%
  mutate(time = time + 1) %>%
  {
    set_names(., ifelse(names(.) == 'geo' | names(.) == 'time', names(.),
                        paste0(names(.), '_1')))
  } %>%
  merge(dat_plm, all.x = F, all.y = T)

dat_plm <- dat_plm %>%
  select(!ends_with("_1")) %>%
  select(-TOTFERRT) %>%
  mutate(time = time + 2) %>%
  {
    set_names(., ifelse(names(.) == 'geo' | names(.) == 'time', names(.),
                        paste0(names(.), '_1_1')))
  } %>%
  merge(dat_plm, all.x = F, all.y = T)

dat_plm <- dat_plm %>%
  select(-TOTFERRT) %>%
  mutate_at(-(1:2), function(x) x^2) %>%
  {
    set_names(., ifelse(names(.) == 'geo' | names(.) == 'time', names(.),
                        paste0(names(.), '_2')))
```

```r
  } %>%
  merge(dat_plm, all.x = F, all.y = T)

## Initial models ==============================================================

library(plm)

m_panels <- c(
  'TOTFERRT ~ edu_index_l_l + health_index_l_l + GDPindex_l_l + FAM_l_l +
  uY15_l_l + uY20_l_l + uY25_l_l + edu_index_l + health_index_l + GDPindex_l + FAM_l +
  uY15_l + uY20_l + uY25_l + edu_index + health_index + GDPindex + FAM + uY15 + uY20 +
  uY25',
  'TOTFERRT ~ edu_index_l_l + health_index_l_l + GDPindex_l_l + FAM_l_l + edu_index_l +
  health_index_l + GDPindex_l + FAM_l + edu_index + health_index + GDPindex + FAM'
) %>%
  lapply(function(formula) {
    pooling <- plm(eval(formula), data = dat_plm, model = "pooling")
    within <- plm(eval(formula), data = dat_plm, model = "within")
    random <- plm(eval(formula), data = dat_plm, model = "random")

    list(
      tests = c(
        pooltest(pooling, within)$p.value,
        phtest(within, random)$p.value,
        plm::r.squared(within, dfcor = T)),
      model = within,
      OLS = lm(data = dat_plm, formula = eval(paste(formula, "+ geo")))
    )
  })
### Plot coefficients ##########################################################

m_panels %>%
  lapply(function(output) {
    output$model %>% broom::tidy(conf.int = T) %>%
      rownames_to_column()
  }) %>%
  reduce(rbind) %>%
  mutate(
    rowname = paste0("Model ", as.roman(cumsum(rowname == 1)), "."),
    term = f.clean_names(term, Tosparse = T),
    term = gsub("_2", "^2", term),
    term = gsub("_l_l", '["t = -2"]', term),
    term = gsub("_l", '["t = -1"]', term),
  ) %>%
  ggplot() +
  aes(estimate, term, color = p.value <= .05) +
  geom_vline(xintercept = 0, color = "gray4") +
  geom_point() +
  geom_pointrange(aes(xmin = conf.low, xmax = conf.high)) +
  facet_wrap(~rowname, nrow = 1) +
  labs(x = "Estimated coefficient", y = "Term", color = "Corresponding p-value \u2264 5%") +
  scale_color_viridis_d(option = "magma", begin = .2, end = .7) +
  scale_y_discrete(labels=scales::parse_format()) +
```

```
  theme(
    legend.position = "bottom",
    legend.direction = "horizontal"
  )

### Model descriptions ##############################################################

m_panels %>%
  lapply(function(output) {
    c(output$tests, nrow(augment(output$OLS)))
  }) %>%
  reduce(rbind) %>%
  t() %>%
  data.frame() %>%
  mutate_all(function(x) c(scales::percent(x[1:3], accuracy = .01),
                           as.character(x[4]))) %>%
  {set_names(., paste0("Model ", as.roman(1:ncol(.)), "."))} %>%
  mutate(
    Indicator = c("Pooltest", "Phtest", "Adjusted $R^2$", "Observations")
  ) %>%
  select(Indicator, everything()) %>%
  knitr::kable(caption = "Models", align = c("l", "c", "c", "c"))

# Framework I. ================================================================

## Bias of framework ##############################################################

names(dat_plm) %>%
  { ifelse(
    . %in% c("geo", "time") | str_detect(., "_1") | str_detect(., "_2"), NA, .
  )} %>%
  na.omit() %>%
  lapply(function(variable){
    x <- pull(dat_plm, variable) %>% na.omit()
    y <- pull(na.omit(dat_plm), variable)
    t <- t.test(x, y)
    tibble(
      Variable = f.clean_names(variable),
      'Mean in total sample' = mean(x),
      'Mean in used sample' = mean(y),
      'Number of observations in the total sample' = length(x),
      'T-statistic' = t$statistic,
      'P-value' = scales::percent(t$p.value, accuracy = .01)
    )
  }
  ) %>% reduce(rbind) %>%
  knitr::kable(caption = 'T-tests', digits = 4,
               align = c('l', 'c', 'c', 'c', 'c', 'c', 'c'))

dat_plm %>%
  na.omit() %>%
  group_by(geo) %>%
  summarise(
```

```r
    values = n()
  ) %>%
  plot_NUTS2(all.x = T)

### Run lasso regression ##########################################################

library(glmnet)

y <- na.omit(dat_plm)$TOTFERRT
X <- model.matrix(TOTFERRT ~ ., data = select(na.omit(dat_plm), -time))
LASSO <- cv.glmnet(X, y)

tidy(LASSO) %>%
  ggplot() +
  aes(log(lambda), ymin = conf.low*1000, ymax = conf.high*1000) +
  geom_line(aes(log(lambda), estimate*1000, color = "Mean-Squared Error")) +
  geom_step(aes(y = nzero, color = "Number of used explanatory variables"),
            size = 1) +
  geom_ribbon(alpha = .4) +
  geom_hline(yintercept = 0, size = 1) +
  geom_vline(aes(xintercept = log(LASSO$lambda.1se), color = "lambda.1se"),
             # TODO name the line
             linetype = 2) +
  geom_vline(aes(xintercept = log(LASSO$lambda.min), color = "lambda.min"),
             linetype = 2) +
  scale_y_continuous(
    name = "Number of used explanatory variables",
    sec.axis = sec_axis( trans=~./1000, name = "Mean-Squared Error")
  ) +
  labs(y = "Mean-Squared Error", x = "Log(\u03BB)", color = NULL) +
  scale_color_viridis_d(option = "magma", end = .8) +
  theme(legend.position = "bottom", legend.direction = "horizontal")

lasso_coefs <- capture.output(
  coef(LASSO, LASSO$lambda.1se)
) %>%
  .[-(1:2)] %>%
  {tibble(x = .)} %>%
  mutate(term = gsub(" .*", "", x), coef = gsub(".* ", "", x)) %>%
  select(-x) %>%
  filter(!str_detect(term, "geo") & coef != "" & term != "(Intercept)")

m_panels2 <- paste("TOTFERRT ~", paste(lasso_coefs$term, collapse = " + ")) %>%
  lapply(function(formula) {
    pooling <- plm(eval(formula), data = dat_plm, model = "pooling")
    within <- plm(eval(formula), data = dat_plm, model = "within")
    random <- plm(eval(formula), data = dat_plm, model = "random")
    list(
      tests = c(
        pooltest(pooling, within)$p.value,
        phtest(within, random)$p.value,
        plm::r.squared(within, dfcor = T)),
      model = within,
```

```r
      OLS = lm(data = dat_plm, formula = eval(paste(formula, "+ geo")))
    )
  }
  )

m_panels2 %>%
  lapply(function(output) {
    output$tests
  })

standard_beta <- m_panels2[[1]]$OLS %>%
  QuantPsyc::lm.beta() %>%
  {tibble(term = names(.), beta = .)} %>%
  filter(!str_detect(term, "geo"))

standard_beta <- augment(m_panels2[[1]]$OLS) %>%
  select(TOTFERRT:uY25) %>%
  cor() %>%
  data.frame() %>%
  select(1) %>%
  rownames_to_column() %>%
  rename(term = rowname) %>%
  merge(standard_beta) %>%
  mutate(explain = abs(TOTFERRT*beta)) %>%
  select(term, cor = TOTFERRT, standard_beta = beta, explain)

lasso_coefs %>%
  rename(lasso = coef) %>%
  merge(tidy(m_panels2[[1]]$OLS, conf.int = T)) %>%
  merge(standard_beta) %>%
  mutate_at(-1, function(x) as.numeric(x)) %>%
  pivot_longer(c(lasso:estimate, cor:explain)) %>%
  mutate(
    conf.low = ifelse(name != "estimate", NA, conf.low),
    conf.high = ifelse(name != "estimate", NA, conf.high),
    lag = ifelse(str_detect(term, "_l"),
                 ifelse(str_detect(term, "_l_l"), 2, 1), 0),
    bar = ifelse(name %in% c("cor", "explain"), value, NA),
    value = ifelse(!(name %in% c("cor", "explain")), value, NA),
    term = f.clean_names(term, Tosparse = T),
    term = gsub("_2", "^2", term),
    term = gsub("_l_l", '["t = -2"]', term),
    term = gsub("_l", '["t = -1"]', term),
    name = case_when(
      name == "cor" ~ 'Correlation coefficient',
      name == "estimate" ~ 'Coefficient in OLS',
      name == "lasso" ~ 'Coefficient in lasso regression',
      name == "standard_beta" ~ 'Standardized coefficient',
      name == "explain" ~ 'Contribution to R-squared'
    ),
    name = factor(
      name, levels = c(
        'Correlation coefficient',
```

```
        'Coefficient in lasso regression',
        'Coefficient in OLS',
        'Standardized coefficient',
        'Contribution to R-squared'
      )
    ),
  ) %>%
  {
    ggplot(.) +
      geom_vline(xintercept = 0) +
      geom_point(aes(value, term, fill = factor(lag)), size = 3) +
      geom_col(aes(bar, term, fill = factor(lag)), color = "black") +
      scale_fill_viridis_d(option = "magma", begin = .3, end = .7) +
      scale_y_discrete(labels=scales::parse_format()) +
      facet_wrap(~ name, scales = "free_x") +
      labs(x = NULL, y = "Term", fill = "Lag") +
      theme(legend.position = "bottom", legend.direction = "horizontal")
  }

dat_plm <- dat_plm %>%
  select(!starts_with("uY"))

LASSO <- cv.glmnet(model.matrix(TOTFERRT ~ ., data = select(na.omit(dat_plm), -time)),
                   na.omit(dat_plm)$TOTFERRT)

names(dat_plm) %>%
  {ifelse(
    . %in% c("geo", "time") | str_detect(., "_1") | str_detect(., "_2"), NA, .
  )} %>%
  na.omit() %>%
  lapply(function(variable){
    x <- pull(dat_plm, variable) %>% na.omit()
    y <- pull(na.omit(dat_plm), variable)
    t <- t.test(x, y)
    tibble(
      Variable = f.clean_names(variable),
      'Mean in total sample' = mean(x),
      'Mean in used sample' = mean(y),
      'Number of observations in total sample' = length(x),
      'T-statistic' = t$statistic,
      'P-value' = scales::percent(t$p.value, accuracy = .01)
    )
  }
  ) %>% reduce(rbind) %>%
  knitr::kable(caption = 'T-tests', digits = 4,
               align = c('l', 'c', 'c', 'c', 'c', 'c', 'c'))

dat_plm %>%
  na.omit() %>%
  group_by(geo) %>%
  summarise(
    values = n()
  ) %>%
```

```r
  plot_NUTS2(all.x = T)

lasso_coefs <- capture.output(
  coef(LASSO, LASSO$lambda.1se)
) %>%
  .[-(1:2)] %>%
  {tibble(x = .)} %>%
  mutate(term = gsub(" .*", "", x), coef = gsub(".* ", "", x)) %>%
  select(-x) %>%
  filter(!str_detect(term, "geo") & coef != "" & term != "(Intercept)")

m_panels2 <- paste("TOTFERRT ~", paste(lasso_coefs$term, collapse = " + ")) %>%
  lapply(function(formula) {
    pooling <- plm(eval(formula), data = dat_plm, model = "pooling")
    within <- plm(eval(formula), data = dat_plm, model = "within")
    random <- plm(eval(formula), data = dat_plm, model = "random")
    list(
      tests = c(
        pooltest(pooling, within)$p.value,
        phtest(within, random)$p.value,
        plm::r.squared(within, dfcor = T)),
      model = within,
      OLS = lm(data = dat_plm, formula = eval(paste(formula, "+ geo")))
    )
  }
  )

m_panels2 %>%
  lapply(function(output) {
    output$tests
  })

standard_beta <- m_panels2[[1]]$OLS %>%
  QuantPsyc::lm.beta() %>%
  {tibble(term = names(.), beta = .)} %>%
  filter(!str_detect(term, "geo"))

standard_beta <- augment(m_panels2[[1]]$OLS) %>%
  select(TOTFERRT:edu_index) %>%
  cor() %>%
  data.frame() %>%
  select(1) %>%
  rownames_to_column() %>%
  rename(term = rowname) %>%
  merge(standard_beta) %>%
  mutate(explain = abs(TOTFERRT*beta)) %>%
  select(term, cor = TOTFERRT, standard_beta = beta, explain)

lasso_coefs %>%
  rename(lasso = coef) %>%
  merge(tidy(m_panels2[[1]]$OLS, conf.int = T)) %>%
  merge(standard_beta) %>%
  mutate_at(-1, function(x) as.numeric(x)) %>%
```

```r
  pivot_longer(c(lasso:estimate, cor:explain)) %>%
  mutate(
    conf.low = ifelse(name != "estimate", NA, conf.low),
    conf.high = ifelse(name != "estimate", NA, conf.high),
    lag = ifelse(str_detect(term, "_l"),
                 ifelse(str_detect(term, "_l_l"), 2, 1), 0),
    bar = ifelse(name %in% c("cor", "explain"), value, NA),
    value = ifelse(!(name %in% c("cor", "explain")), value, NA),
    term = f.clean_names(term, Tosparse = T),
    term = gsub("_2", "^2", term),
    term = gsub("_l_l", '["t = -2"]', term),
    term = gsub("_l", '["t = -1"]', term),
    name = case_when(
      name == "cor" ~ 'Correlation coefficient',
      name == "estimate" ~ 'Coefficient in OLS',
      name == "lasso" ~ 'Coefficient in lasso regression',
      name == "standard_beta" ~ 'Standardized coefficient',
      name == "explain" ~ 'Contribution to R-squared'
    ),
    name = factor(
      name, levels = c(
        'Correlation coefficient',
        'Coefficient in lasso regression',
        'Coefficient in OLS',
        'Standardized coefficient',
        'Contribution to R-squared'
      )
    )
  ) %>%
  {
    ggplot(.) +
      geom_vline(xintercept = 0) +
      geom_point(aes(value, term, fill = factor(lag)), size = 3) +
      geom_col(aes(bar, term, fill = factor(lag)), color = "black") +
      scale_fill_viridis_d(option = "magma", begin = .3, end = .7) +
      scale_y_discrete(labels=scales::parse_format()) +
      facet_wrap(~ name, scales = "free_x") +
      labs(x = NULL, y = "Term", fill = "Lag") +
      theme(legend.position = "bottom", legend.direction = "horizontal")
  }
```