01_script_master_women_exchange_replication.R

js4618

2023-06-06

```
######################################################################
##                          Preliminaries                           ##
######################################################################

# Load useful packages  ------------------------------------------------------

# install.packages("pacman") # Uncomment if `pacman` package is not already
# installed
library(pacman)

p_load(
tidyverse,
janitor,
broom,
countrycode,

# Modeling
lmtest,
plm,
tscsdep,
spdep,
spatialreg,
urca,
dynamac,
fixest,
bayestestR,

# Graphs and Figures
patchwork,
marginaleffects,

# Tables
texreg,
modelsummary,
kableExtra,

# Project managements
here
)

# Tell R where you are
i_am("01_script_master_women_exchange_replication.R")
```

```
## here() starts at /Users/js4618/Desktop/replication_women_exchange
# Load useful functions  ------------------------------------------


# Create a function that displays coefficient estimates with PCSEs
show_pcse <- function (x) {
  #'  Show Beck and Katz (1995) panel-corrected standard errors standard errors
  #'  for time-series cross-section linear models. Note that this code will
  #'  *only* work with models estimated with the `plm` package.

  #' @param x:  A `plm` linear model object.
  #'
  #' @return The `summary.plm` method for plm objects but with the Beck and
  #' Katz (1005) standard errors and associated t-values and p-values.
  #'
  #' @example show_pcse(x)

  summary(x, vcovBK(x, type = "HC1", cluster = "time"))
}

# I use `texreg` to make the tables. When using "robust" standard errors,
# `texreg` requires functions to extract the correct standard errors and
# p-values. The two functions extract the relevant items.
extract_pcse <- function(x) {
  #' Extract the Beck and Katz (1995) standard errors for use in `texreg`
  #' tables. Note that this code will *only* work with model estimated with the
  #' `plm` package.

  #' @argument x: A plm linear model object.

  #' @returns When used inside the `texreg` function with the standard error
  #'   specified with the `override.se` list, this function allows the user to
  #'   substitute OLS standard errors with those recommended by Beck and Katz
  #'   (1995).
  #'
  #' @example texreg(x, override.se = extract_pcse(x))

  coeftest(x, vcov. = vcovBK(x, type = "HC1", cluster = "time"))[, 2]

}

extract_pcse_sig <- function(x) {
  #' Extract the p-value associated with the Beck and Katz (1995) standard
  #' errors for use in `texreg` tables Note that this code will *only* work
  #' with model estimated with the `plm` package.

  #' @argument x: A plm linear model object.

  #' @returns When used inside the `texreg` function with the p-values specified
  #'   with the `override.pvalue` list, this function allows the user to
  #'   substitute the "stars" for OLS standard errors with those commensurate
  #'   with the Beck and Katz (1995) standard errors.
  #'
```

```
  #'   @example texreg(x, override.se = extract_pcse(x), override.pvalues = extract_pcse_sig(x))

  coeftest(x, vcov. = vcovBK(x, type = "HC1", cluster = "time"))[, 4]
}

# One model in the paper uses the `dynardl` package, but `texreg` doesn't
# support those objects out of the box. This can be fixed by adding a custom
# `extract` function that grabs all the relevant information.
extract.dynardl <- function(model,
                            include.rsquared = FALSE,
                            include.adj.rsquared = FALSE,
                            include.nobs = TRUE) {
  s <- summary(model)
  names <- rownames(s$coefficients)
  co <- s$coefficients[, 1]
  se <- s$coefficients[, 2]
  pval <- s$coefficients[, 4]

  gof <- numeric()
  gof.names <- character()
  gof.decimal <- logical()

  if (include.rsquared == TRUE) {
    rs <- s$r.squared
    gof <- c(gof, rs)
    gof.names <- c(gof.names, "R$^2$")
    gof.decimal <- c(gof.decimal, TRUE)
  }

  if (include.adj.rsquared == TRUE) {
    adj <- s$adj.rsquared
    gof <- c(gof, adj)
    gof.names <- c(gof.names, "Adj.\\ R$^2$")
    gof.decimal <- c(gof.decimal, TRUE)
  }

  if (include.nobs == TRUE) {
    n <- nrow(model$model$model)
    gof <- c(gof, n)
    gof.names <- c(gof.names, "Num.\\ obs.")
    gof.decimal <- c(gof.decimal, TRUE)
  }

    tr <- createTexreg(
    coef.names = names,
    coef = co,
    se = se,
    pvalues = pval,
    gof.names = gof.names,
    gof = gof

  )
  return(tr)
```

```
}

# Now, we register the function as a method for the generic extract() function
setMethod("extract",
          signature = className("dynardl", "dynardl"),
          definition = extract.dynardl)

# Load useful settings for tables ----------------------------------------

# Make list of variable names for `texreg` tables
list_var_names_texreg <- list(
  'flfplag'                          = 'LDV$_{(t-1)}$',
  'flfp_mlfplag'                     = 'LDV$_{(t-1)}$',
  'dflfplag'                         = '$\\Delta$LDV$_{(t-1)}$',
  'dflfp_mlfplag'                    = '$\\Delta$LDV$_{(t-1)}$',
  'log_wdi_overvaluedlag'            = 'logOvervalued$_{(t-1)}$',
  'dlog_wdi_overvalued'              = '$\\Delta$logOvervalued$_t$',
  'log_gdpcaplag'                    = 'GDP$_{(t-1)}$',
  'dlog_gdpcap'                      = '$\\Delta$GDP$_t$',
  'sqlog_gdpcaplag'                  = 'GDP$^{2}_{(t-1)}$',
  'dsqlog_gdpcap'                    = '$\\Delta$GDP$^{2}_t$',
  'log_oilgascaplag'                 = 'Resource Rents$_{(t-1)}$',
  'dlog_oilgascap'                   = '$\\Delta$Resource Rents$_t$',
  'wdi_fertilitylag'                 = 'Fertility$_{(t-1)}$',
  'dwdi_fertility'                   = '$\\Delta$Fertility$_t$',
  'vdem_elec_demlag'                 = 'Regime Type$_{(t-1)}$',
  'dvdem_elec_dem'                   = '$\\Delta$Regime Type$_t$',
  'as_factor_irr_rate_regimelag2'    = 'Exchange Rate: Narrow Crawling$_t$',
  'as_factor_irr_rate_regimelag3'    = 'Exchange Rate: Wide Crawling$_t$',
  'as_factor_irr_rate_regimelag4'    = 'Exchange Rate: Freely Floating$_t$',
  'as_factor_irr_rate_regimelag5'    = 'Exchange Rate: Freely Falling$_t$',
  'as_factor_irr_rate_regimelag6'    = 'Exchange Rate: Dual Market$_t$',
  'rho'                              = 'Spatial Lag: $\\Delta$Dependent Variable$_t$',
  '$\\rho$'                          = 'Spatial Lag: $\\Delta$Dependent Variable$_t$'
)


# Load useful settings for figures ----------------------------------------

# Figures
theme_set(theme_minimal(base_size = 10))
theme_update(text = element_text(family = "sans"))


# Load baseline data ------------------------------------------------------

# TSCS data frame spanning 1990--2015 inclusive
df_tscs <-
  read_csv("05_data_women_exchange_tscs_replication.csv") %>%
  group_by(cowcode) %>%
  # Create lags and differenced variables
  mutate(dflfplag = dplyr::lag(dflfp, 1),
         flfplag2 = dplyr::lag(flfp, 2),
```

```
        dflfp_mlfplag = dplyr::lag(dflfp_mlfp, 1)) %>%
  ungroup()
```

```
## Rows: 5044 Columns: 32
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): country, region
## dbl (30): cowcode, year, oecd, log_wdi_overvalued, log_wdi_overvaluedlag, dlog_wdi_overvalued, flfp,
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# The spatial regressions need alternative names for the some countries to
# accurately create k-nearest neighbors. Fix those names here.
df_tscs  <- mutate(df_tscs, country_space = country)

df_tscs$country_space <- recode_factor(df_tscs$country_space, "Bahamas, The"="Bahamas")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Belarus"="Belarus (Byelorussia)")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Bosnia and Herzegovina"="Bosnia-Herzegovi
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Brunei Darussalam"="Brunei")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Burkina Faso"="Burkina Faso (Upper Volta)
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Cabo Verde"="Cape Verde")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Cambodia"="Cambodia (Kampuchea)")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Congo, Dem. Rep."="Congo, Democratic Repu
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Congo, Rep."="Congo")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Cote d'Ivoire"="Cote D'Ivoire")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Egypt, Arab Rep."="Egypt")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Ethiopia -pre 1993"="Ethiopia")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Ethiopia 1993-"="Ethiopia")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Gambia, The"="Gambia")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Germany"="German Federal Republic")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Iran, Islamic Rep."="Iran (Persia)")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Italy"="Italy/Sardinia")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Korea, Rep."="Korea, Republic of")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Lao PDR"="Laos")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Macedonia, FYR"="Macedonia (FYROM/North M
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Madagascar"="Madagascar (Malagasy)")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Myanmar"="Myanmar (Burma)")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Pakistan-post-1972"="Pakistan")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Pakistan-pre-1972"="Pakistan")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Romania"="Rumania")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Russian Federation"="Russia (Soviet Union
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Slovak Republic"="Slovakia")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Sri Lanka"="Sri Lanka (Ceylon)")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Eswatini"="Swaziland (Eswatini)")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Suriname"="Surinam")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Syrian Arab Republic"="Syria")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Tanzania"="Tanzania (Tanganyika)")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Turkey"="Turkey (Ottoman Empire)")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "United States"="United States of America
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Venezuela, RB"="Venezuela")
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Vietnam"="Vietnam, Democratic Republic of
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Yemen, Rep."="Yemen (Arab Republic of Yem
df_tscs$country_space <- recode_factor(df_tscs$country_space, "Zimbabwe"="Zimbabwe (Rhodesia)")
```

```r
# Load ILO industry employment data ----------------------------------------------------------

source("02_script_make_ILO_data.R")
```

```
## Rows: 999330 Columns: 12
## -- Column specification --------------------------------------------------------------------
## Delimiter: ","
## chr (10): ref_area.label, indicator.label, source.label, sex.label, classif1.label, classif2.label, 
## dbl  (2): time, obs_value
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
######################################################################
##                  Table 1: Summary Statistics                     ##
######################################################################


# Table 1: Summary Statistics ----------------------------------------

# Grab the data in the baseline models to extract for summary statistics
df_sum_stats <-
  plm(
    dflfp ~ flfplag + plm::lag(dflfp, 1) +
      log_wdi_overvaluedlag + dlog_wdi_overvalued +
      log_gdpcaplag + dlog_gdpcap + sqlog_gdpcaplag +
      dsqlog_gdpcap + log_oilgascaplag + dlog_oilgascap - country - year,
    data = subset(df_tscs, oecd == 0 & country != "Zimbabwe"),
    model = "within",
    effect = "individual",
    index = c("cowcode", "year")
  )$model

# Order the variables
df_sum_stats <- df_sum_stats |>
  select(!contains("plm")) |>
  rename(
    "D.FLFP" = dflfp,
    "L.FLFP" = flfplag,
    "L.REER Overvaluation" = log_wdi_overvaluedlag,
    "D.REER Overvaluation" = dlog_wdi_overvalued,
    "L.Per Capita GDP" = log_gdpcaplag,
    "D.Per Capita GDP" = dlog_gdpcap,
    "L.Squared Per Capita GDP" = sqlog_gdpcaplag,
    "D.Squared Per Capita GDP" = dsqlog_gdpcap,
    "L.Resource Rents" = log_oilgascaplag,
    "D.Resource Rents" = dlog_oilgascap
  )

# View the table
datasummary(
  All(df_sum_stats) ~ N + Mean + SD + Min + Median + Max,
```

```
  data = df_sum_stats,
  output = "markdown",
  title = "Summary Statistics",
  notes = "L.X and D.X refer to the lagged and differenced values of X."
)
```

Table: Summary Statistics

	N	Mean	SD	Min	Median	Max
D.FLFP	3245	0.14	0.83	-6.39	0.08	6.14
L.FLFP	3245	50.48	16.97	6.08	49.95	90.35
L.REER Overvaluation	3245	0.00	0.96	-9.13	-0.01	7.12
D.REER Overvaluation	3245	-0.01	0.18	-4.72	0.00	2.13
L.Per Capita GDP	3245	7.89	1.31	4.75	7.94	11.19
D.Per Capita GDP	3245	0.02	0.06	-0.97	0.02	0.88
L.Squared Per Capita GDP	3245	64.03	21.00	22.58	63.05	125.30
D.Squared Per Capita GDP	3245	0.35	0.88	-17.36	0.38	13.48
L.Resource Rents	3245	2.28	2.85	0.00	0.47	10.24
D.Resource Rents	3245	0.01	0.32	-2.47	0.00	5.28

Note: ^^ L.X and D.X refer to the lagged and differenced values of X.

```
######################################################################
##                    Table 2: Baseline Models                      ##
######################################################################


# Table 2: FLFP ~ REER  + Country Effects ----------------------------------------------------

m_flfp_base <- plm(
  dflfp ~ flfplag + dflfplag +
    log_wdi_overvaluedlag + dlog_wdi_overvalued +
    log_gdpcaplag + dlog_gdpcap +
    sqlog_gdpcaplag + dsqlog_gdpcap +
    log_oilgascaplag + dlog_oilgascap,
  data = subset(df_tscs,
                oecd == 0 &
                  country != "Zimbabwe"),
  model = "within",
  effect = "individual",
  index = c("cowcode", "year")
)

# View results with PCSEs
show_pcse(m_flfp_base)
```

```
## Oneway (individual) effect Within Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovBK(x, type = "HC1", cluster = "time")
##
## Call:
## plm(formula = dflfp ~ flfplag + dflfplag + log_wdi_overvaluedlag +
##     dlog_wdi_overvalued + log_gdpcaplag + dlog_gdpcap + sqlog_gdpcaplag +
```

```
##     dsqlog_gdpcap + log_oilgascaplag + dlog_oilgascap, data = subset(df_tscs,
##     oecd == 0 & country != "Zimbabwe"), effect = "individual",
##     model = "within", index = c("cowcode", "year"))
##
## Unbalanced Panel: n = 147, T = 5-24, N = 3245
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -6.5269712 -0.2033263  0.0048274  0.2116495  6.4088103
##
## Coefficients:
##                          Estimate Std. Error t-value  Pr(>|t|)
## flfplag                 -0.048426   0.011129 -4.3516 1.396e-05 ***
## dflfplag                 0.151854   0.056170  2.7035 0.0068992 **
## log_wdi_overvaluedlag -0.118229   0.035552 -3.3255 0.0008929 ***
## dlog_wdi_overvalued   -0.149872   0.053392 -2.8070 0.0050314 **
## log_gdpcaplag         -1.433910   0.343414 -4.1755 3.056e-05 ***
## dlog_gdpcap           -2.229163   0.905063 -2.4630 0.0138325 *
## sqlog_gdpcaplag        0.101055   0.022521  4.4871 7.483e-06 ***
## dsqlog_gdpcap          0.184915   0.066242  2.7915 0.0052788 **
## log_oilgascaplag      -0.014387   0.017340 -0.8297 0.4067880
## dlog_oilgascap         0.046859   0.038711  1.2105 0.2261871
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    1867
## Residual Sum of Squares: 1760.6
## R-Squared:      0.056984
## Adj. R-Squared: 0.0093446
## F-statistic: 6.99925 on 10 and 23 DF, p-value: 5.9489e-05
```

```
# Sanity check that my shorthand function returns the same estimates as the
# coeftest function
coeftest(m_flfp_base,
         vcov. = vcovBK(m_flfp_base, type = "HC1", cluster = "time"))
```

```
##
## t test of coefficients:
##
##                          Estimate Std. Error t value  Pr(>|t|)
## flfplag                 -0.048426   0.011129 -4.3516 1.396e-05 ***
## dflfplag                 0.151854   0.056170  2.7035 0.0068992 **
## log_wdi_overvaluedlag -0.118229   0.035552 -3.3255 0.0008929 ***
## dlog_wdi_overvalued   -0.149872   0.053392 -2.8070 0.0050314 **
## log_gdpcaplag         -1.433910   0.343414 -4.1755 3.056e-05 ***
## dlog_gdpcap           -2.229163   0.905063 -2.4630 0.0138325 *
## sqlog_gdpcaplag        0.101055   0.022521  4.4871 7.483e-06 ***
## dsqlog_gdpcap          0.184915   0.066242  2.7915 0.0052788 **
## log_oilgascaplag      -0.014387   0.017340 -0.8297 0.4067880
## dlog_oilgascap         0.046859   0.038711  1.2105 0.2261871
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Table 2: FLFP ~ REER  + Extended Controls ------------------------------
```

```r
# Create a dataset to estimate the model
df <- df_tscs %>%
  filter(
    oecd == 0 &
      country_space != "Zimbabwe (Rhodesia)" &
      country_space != "Tonga" # Causes the nearest neighbor function to balk
  )

# Estimate the baseline model using `lm` instead of `plm`
pre <- lm(
  dflfp ~ flfplag + dflfplag +
    log_wdi_overvaluedlag + dlog_wdi_overvalued +
    log_gdpcaplag + dlog_gdpcap +
    sqlog_gdpcaplag + dsqlog_gdpcap +
    log_oilgascaplag + dlog_oilgascap +
    as_factor(irr_rate_regimelag) +
    wdi_fertilitylag + dwdi_fertility +
    vdem_elec_demlag + dvdem_elec_dem +
    as_factor(cowcode) - cowcode - year - country_space,
  data = df
)

# Take the data used in the estimation and put it in a data frame with tidy
# names
df2 <- pre$model %>% data.frame() %>% clean_names()

# Make the spatial weights where k = 10
W <- make_ntspmat(pre, ci = country_space, y = year, k = 10)
```

```
## [1] 1992
##  [1] 42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290 310
## [38] 438 439 450 451 452 461 471 475 481 482 483 484 490 500 501 510 516 517 530 540 541 551 553 560
## [75] 663 666 670 679 710 712 732 750 760 770 771 780 790 800 812 820 830 840 850 910 935 940 950
## [1] 1993
##   [1] 42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290 310
##  [38] 436 437 438 439 450 451 452 461 471 475 481 482 483 484 490 500 501 510 516 517 530 540 541 551
##  [75] 651 660 663 666 670 679 702 710 712 732 750 760 770 771 780 790 800 812 820 830 840 850 910 935
## [1] All of your Countries are Matched.
## [1] 1994
##   [1] 42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290 310
##  [38] 433 434 435 436 437 438 439 450 451 452 461 471 475 481 482 483 484 490 500 501 510 516 517 530
##  [75] 615 616 630 640 651 660 663 666 670 679 702 710 712 732 750 760 770 771 780 790 800 811 812 820
## [1] All of your Countries are Matched.
## [1] 1995
##   [1] 42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290 310
##  [38] 411 420 432 433 434 435 436 437 438 439 450 451 452 461 471 475 481 482 483 484 490 500 501 510
##  [75] 581 590 600 615 616 630 640 651 660 663 666 670 679 701 702 703 705 710 712 732 750 760 770 771
## [1] All of your Countries are Matched.
## [1] 1996
##   [1] 42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290 310
##  [38] 369 370 371 373 402 404 411 420 432 433 434 435 436 437 438 439 450 451 452 461 471 475 481 482
##  [75] 560 565 570 571 572 580 581 590 600 615 616 630 640 651 660 663 666 670 679 690 701 702 703 704
## [112] 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
```

```
## [1] 1997
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290
##  [38] 367 368 369 370 371 372 373 402 404 411 420 432 433 434 435 436 437 438 439 450 451 452 461 471
##  [75] 541 551 553 560 565 570 571 572 580 581 590 600 615 616 625 630 640 651 660 663 666 670 679 690
## [112] 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 1998
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290
##  [38] 366 367 368 369 370 371 372 373 402 404 411 420 432 433 434 435 436 437 438 439 450 451 452 461
##  [75] 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 625 630 640 651 660 663 666 670
## [112] 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 1999
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290
##  [38] 366 367 368 369 370 371 372 373 402 404 411 420 432 433 434 435 436 437 438 439 450 451 452 461
##  [75] 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 625 630 640 651 660 663 666 670
## [112] 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2000
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290
##  [38] 366 367 368 369 370 371 372 373 402 404 420 432 433 434 435 436 437 438 439 450 451 452 461 471
##  [75] 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 651 660 663 666 670
## [112] 771 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2001
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290
##  [38] 366 367 368 369 370 371 372 373 402 404 420 432 433 434 435 436 437 438 439 450 451 452 461 471
##  [75] 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 651 660 663 666 670
## [112] 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2002
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290
##  [38] 366 367 368 369 370 371 372 373 402 403 404 420 432 433 434 435 436 437 438 439 450 451 452 461
##  [75] 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 651 660 663 660
## [112] 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2003
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 365 366 367 368 369 370 371 372 373 402 403 404 420 432 433 434 435 436 437 438 439 450 451 451
##  [75] 530 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 651 660 663
## [112] 750 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2004
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 365 366 367 368 369 370 371 372 373 402 403 404 420 432 433 434 436 437 438 439 450 451 452 461
##  [75] 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 651 660 663 660
## [112] 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2005
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 365 366 367 368 369 370 371 372 373 402 403 404 420 432 433 434 436 437 438 439 450 451 452 461
##  [75] 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 651 660 663
## [112] 750 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
```

```
## [1] 2006
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 365 366 367 368 369 370 371 372 373 402 403 404 420 432 433 434 435 436 437 438 439 450 451 452
##  [75] 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 651 660 663
## [112] 750 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2007
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 360 365 366 367 368 369 370 371 372 373 402 403 404 411 420 432 433 434 435 436 437 438 439 450
##  [75] 517 530 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 651
## [112] 712 732 750 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2008
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 365 366 367 368 369 370 371 372 373 402 403 404 411 420 432 433 434 435 436 437 438 439 450 451
##  [75] 517 530 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 651
## [112] 712 732 750 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2009
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 366 367 368 369 370 371 372 373 402 403 404 411 420 432 433 434 435 436 437 438 439 450 451 452
##  [75] 530 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 651 660
## [112] 732 750 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2010
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 366 367 368 369 370 371 372 373 402 403 404 411 420 432 433 434 435 436 437 438 439 450 451 461
##  [75] 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 651 660 663
## [112] 750 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2011
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 368 369 370 371 372 373 402 403 404 411 420 432 433 434 435 436 437 438 439 450 451 461 471
##  [75] 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 651 660 663 666
## [112] 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2012
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 368 369 370 371 372 373 402 403 404 411 420 432 433 434 435 436 437 438 439 450 451 461 471
##  [75] 541 551 553 560 565 570 571 572 580 581 590 600 615 616 625 630 640 645 651 660 663 666 670 679
## [112] 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2013
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 368 369 370 371 372 373 402 403 404 411 420 432 433 434 435 436 437 438 439 450 451 461 471
##  [75] 541 551 553 560 565 570 571 572 580 581 590 600 615 616 625 630 640 645 651 660 663 666 670 679
## [112] 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2014
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 140 145 150 155 160 165 290
##  [38] 369 370 371 372 373 402 403 404 411 420 432 433 434 435 437 438 439 450 451 452 461 471 475 481
##  [75] 553 560 565 570 571 572 580 581 590 600 615 616 625 630 640 645 651 660 663 666 670 679 690 694
## [112] 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
```

```
## [1] 2015
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 110 115 130 140 145 150 155 160 165 290 310
##  [38] 371 372 373 402 403 404 411 420 432 433 434 435 437 438 439 450 451 452 461 471 475 481 482 483
##  [75] 565 570 571 572 580 581 590 600 615 616 625 630 640 645 651 660 663 666 670 679 690 694 698 700
## [112] 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
```

```
# Now, we can estimate the SAR model (may take a couple of minutes to run)
m_flfp_sar <- lagsarlm(
  dflfp ~ flfplag + dflfplag +
    log_wdi_overvaluedlag + dlog_wdi_overvalued +
    log_gdpcaplag + dlog_gdpcap +
    sqlog_gdpcaplag + dsqlog_gdpcap +
    log_oilgascaplag + dlog_oilgascap +
    as_factor_irr_rate_regimelag +
    wdi_fertilitylag + dwdi_fertility +
    vdem_elec_demlag + dvdem_elec_dem +
    as_factor(cowcode),
  listw = mat2listw(W[[2]], style = "W"),
  Durbin = FALSE,
  data = df2
)

# View the results
summary(m_flfp_sar)
```

```
##
## Call:lagsarlm(formula = dflfp ~ flfplag + dflfplag + log_wdi_overvaluedlag +     dlog_wdi_overvalued
##     dsqlog_gdpcap + log_oilgascaplag + dlog_oilgascap + as_factor_irr_rate_regimelag +
##     wdi_fertilitylag + dwdi_fertility + vdem_elec_demlag + dvdem_elec_dem +     as_factor(cowcode), d
##     style = "W"), Durbin = FALSE)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -6.626063 -0.214878  0.006748  0.217916  6.494344
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##                                  Estimate  Std. Error z value  Pr(>|z|)
## (Intercept)                    8.99807983  2.03312955  4.4257 9.612e-06
## flfplag                       -0.05127090  0.00521284 -9.8355 < 2.2e-16
## dflfplag                       0.12059333  0.01859410  6.4856 8.840e-11
## log_wdi_overvaluedlag         -0.11522480  0.05348353 -2.1544 0.0312090
## dlog_wdi_overvalued           -0.15607686  0.08161995 -1.9122 0.0558455
## log_gdpcaplag                 -1.50183416  0.48074217 -3.1240 0.0017842
## dlog_gdpcap                   -2.76978844  1.34691099 -2.0564 0.0397439
## sqlog_gdpcaplag                0.10726443  0.03000672  3.5747 0.0003507
## dsqlog_gdpcap                  0.23014146  0.09055605  2.5414 0.0110401
## log_oilgascaplag              -0.03296244  0.02882514 -1.1435 0.2528183
## dlog_oilgascap                 0.02697239  0.04723406  0.5710 0.5679747
## as_factor_irr_rate_regimelag2 -0.09026717  0.06549040 -1.3783 0.1681024
## as_factor_irr_rate_regimelag3 -0.04968176  0.07519574 -0.6607 0.5088053
## as_factor_irr_rate_regimelag4 -0.00209125  0.15682858 -0.0133 0.9893608
## as_factor_irr_rate_regimelag5  0.02521380  0.08296712  0.3039 0.7612032
## as_factor_irr_rate_regimelag6 -0.13501192  0.17673093 -0.7639 0.4449027
```

```
## wdi_fertilitylag             -0.05251930  0.04061321 -1.2932 0.1959565
## dwdi_fertility               -0.02025214  0.35091851 -0.0577 0.9539782
## vdem_elec_demlag             -0.50814159  0.18741022 -2.7114 0.0067002
## dvdem_elec_dem                0.06146044  0.33118075  0.1856 0.8527743
## as_factor(cowcode)42         -0.61730169  0.27576098 -2.2385 0.0251859
## as_factor(cowcode)51         -0.53854517  0.27470540 -1.9604 0.0499437
## as_factor(cowcode)52         -0.91580866  0.33816846 -2.7081 0.0067661
## as_factor(cowcode)53         -0.53417965  0.31740387 -1.6830 0.0923818
## as_factor(cowcode)70         -1.07494079  0.32613946 -3.2960 0.0009809
## as_factor(cowcode)90         -1.04842704  0.27353659 -3.8329 0.0001267
## as_factor(cowcode)91         -0.78836932  0.25945415 -3.0386 0.0023770
## as_factor(cowcode)92         -0.98508979  0.27378024 -3.5981 0.0003205
## as_factor(cowcode)93         -0.66590088  0.25773539 -2.5837 0.0097758
## as_factor(cowcode)94         -0.85083313  0.31124058 -2.7337 0.0062630
## as_factor(cowcode)95         -0.84194255  0.30083740 -2.7987 0.0051315
## as_factor(cowcode)100        -0.15716660  0.30215606 -0.5202 0.6029587
## as_factor(cowcode)101        -0.93908874  0.34113090 -2.7529 0.0059075
## as_factor(cowcode)110        -1.38998654  0.27963178 -4.9708 6.669e-07
## as_factor(cowcode)115        -1.17656070  0.33243829 -3.5392 0.0004014
## as_factor(cowcode)130        -0.63319389  0.33517436 -1.8891 0.0588720
## as_factor(cowcode)135         0.74076366  0.27477822  2.6959 0.0070207
## as_factor(cowcode)140        -0.56258046  0.31131146 -1.8071 0.0707419
## as_factor(cowcode)145        -0.08648729  0.26787307 -0.3229 0.7467962
## as_factor(cowcode)150        -0.38902851  0.25988953 -1.4969 0.1344194
## as_factor(cowcode)155        -0.99148219  0.32975908 -3.0067 0.0026411
## as_factor(cowcode)160        -0.88828043  0.30993003 -2.8661 0.0041561
## as_factor(cowcode)165        -0.62869908  0.31488282 -1.9966 0.0458673
## as_factor(cowcode)290        -1.23817581  0.30999107 -3.9942 6.490e-05
## as_factor(cowcode)310        -1.46026800  0.32248784 -4.5281 5.951e-06
## as_factor(cowcode)316        -1.33623919  0.33901044 -3.9416 8.094e-05
## as_factor(cowcode)317        -1.49770383  0.35595891 -4.2075 2.582e-05
## as_factor(cowcode)339        -1.07129428  0.27004172 -3.9671 7.274e-05
## as_factor(cowcode)341        -1.63488251  0.36431871 -4.4875 7.206e-06
## as_factor(cowcode)343        -1.36418160  0.29130287 -4.6830 2.827e-06
## as_factor(cowcode)344        -1.40475997  0.33044867 -4.2511 2.128e-05
## as_factor(cowcode)345        -1.29511063  0.35134643 -3.6861 0.0002277
## as_factor(cowcode)346        -1.93591491  0.31660420 -6.1146 9.679e-10
## as_factor(cowcode)349        -1.87934896  0.44278774 -4.2444 2.192e-05
## as_factor(cowcode)352        -1.18298830  0.40800997 -2.8994 0.0037387
## as_factor(cowcode)355        -1.30257942  0.28585848 -4.5567 5.196e-06
## as_factor(cowcode)359        -1.36412378  0.27309876 -4.9950 5.884e-07
## as_factor(cowcode)360        -1.40209715  0.29216411 -4.7990 1.595e-06
## as_factor(cowcode)365        -0.70136593  0.32537012 -2.1556 0.0311154
## as_factor(cowcode)366        -1.28600800  0.34214294 -3.7587 0.0001708
## as_factor(cowcode)367        -0.94183259  0.32654511 -2.8842 0.0039237
## as_factor(cowcode)368        -1.11789265  0.30949117 -3.6120 0.0003038
## as_factor(cowcode)369        -1.06003236  0.28195521 -3.7596 0.0001702
## as_factor(cowcode)370        -0.61164866  0.28494676 -2.1465 0.0318302
## as_factor(cowcode)371        -0.90870739  0.26336481 -3.4504 0.0005598
## as_factor(cowcode)372        -0.49579434  0.26435944 -1.8755 0.0607301
## as_factor(cowcode)373        -0.17523687  0.30205724 -0.5801 0.5618171
## as_factor(cowcode)402        -0.71748068  0.27896344 -2.5720 0.0101127
## as_factor(cowcode)403        -0.09135572  0.46908826 -0.1948 0.8455874
## as_factor(cowcode)404         0.03045408  0.25397934  0.1199 0.9045563
```

```
## as_factor(cowcode)411        -0.79622866  0.35021297 -2.2736 0.0229927
## as_factor(cowcode)420        -0.58552805  0.25049270 -2.3375 0.0194129
## as_factor(cowcode)432        -0.17072188  0.28554286 -0.5979 0.5499165
## as_factor(cowcode)433        -0.84151556  0.28230453 -2.9809 0.0028742
## as_factor(cowcode)434         0.58044449  0.26495944  2.1907 0.0284741
## as_factor(cowcode)435        -1.31041694  0.30821852 -4.2516 2.123e-05
## as_factor(cowcode)436         0.25751648  0.29874231  0.8620 0.3886864
## as_factor(cowcode)437        -0.67134182  0.27783872 -2.4163 0.0156791
## as_factor(cowcode)438        -0.00086092  0.24810409 -0.0035 0.9972314
## as_factor(cowcode)439        -0.41038116  0.26996352 -1.5201 0.1284769
## as_factor(cowcode)450        -1.05814761  0.36244654 -2.9195 0.0035064
## as_factor(cowcode)451        -0.29482460  0.25260042 -1.1672 0.2431466
## as_factor(cowcode)452         0.66534032  0.26672376  2.4945 0.0126138
## as_factor(cowcode)461         0.58062956  0.27159470  2.1379 0.0325287
## as_factor(cowcode)471         0.48657452  0.29526432  1.6479 0.0993673
## as_factor(cowcode)475        -0.49257660  0.31088377 -1.5844 0.1130937
## as_factor(cowcode)481        -1.43325950  0.36466123 -3.9304 8.481e-05
## as_factor(cowcode)482        -0.09273247  0.25377929 -0.3654 0.7148084
## as_factor(cowcode)483         0.12157696  0.28969398  0.4197 0.6747238
## as_factor(cowcode)484         0.28660726  0.32534968  0.8809 0.3783608
## as_factor(cowcode)490         0.34006491  0.27801777  1.2232 0.2212629
## as_factor(cowcode)500         0.16741790  0.25926407  0.6457 0.5184460
## as_factor(cowcode)501        -0.12093427  0.24499100 -0.4936 0.6215693
## as_factor(cowcode)510         0.89137535  0.27677072  3.2206 0.0012791
## as_factor(cowcode)516         0.50659884  0.30737203  1.6482 0.0993195
## as_factor(cowcode)517         1.00608021  0.28536810  3.5256 0.0004226
## as_factor(cowcode)530         0.63512648  0.28052149  2.2641 0.0235685
## as_factor(cowcode)531         0.33260197  0.29556026  1.1253 0.2604504
## as_factor(cowcode)540         0.70194711  0.35139008  1.9976 0.0457569
## as_factor(cowcode)541         0.97642078  0.29526209  3.3070 0.0009431
## as_factor(cowcode)551         0.41013191  0.27513663  1.4906 0.1360539
## as_factor(cowcode)553         0.38638617  0.26256455  1.4716 0.1411329
## as_factor(cowcode)560        -0.99883539  0.31262790 -3.1950 0.0013985
## as_factor(cowcode)565        -0.46075488  0.28670767 -1.6071 0.1080424
## as_factor(cowcode)570        -0.27459116  0.24597230 -1.1163 0.2642724
## as_factor(cowcode)571        -0.09529035  0.28107576 -0.3390 0.7345945
## as_factor(cowcode)572        -1.46406104  0.29759225 -4.9197 8.668e-07
## as_factor(cowcode)580         0.99073347  0.27825413  3.5605 0.0003701
## as_factor(cowcode)581        -1.37833938  0.27952752 -4.9310 8.183e-07
## as_factor(cowcode)590        -1.13835602  0.30441442 -3.7395 0.0001844
## as_factor(cowcode)600        -2.15476161  0.31153512 -6.9166 4.626e-12
## as_factor(cowcode)615        -2.55963907  0.38628339 -6.6263 3.441e-11
## as_factor(cowcode)616        -2.11006179  0.33762056 -6.2498 4.110e-10
## as_factor(cowcode)620        -2.20875902  0.42253190 -5.2274 1.719e-07
## as_factor(cowcode)625        -2.14007457  0.33408351 -6.4058 1.496e-10
## as_factor(cowcode)630        -2.59378632  0.40454881 -6.4116 1.440e-10
## as_factor(cowcode)640        -2.39861804  0.35090394 -6.8355 8.170e-12
## as_factor(cowcode)645        -2.02741628  0.43049590 -4.7095 2.483e-06
## as_factor(cowcode)651        -2.24645930  0.34528861 -6.5060 7.716e-11
## as_factor(cowcode)660        -2.58561453  0.36323108 -7.1184 1.092e-12
## as_factor(cowcode)663        -2.76087040  0.35836946 -7.7040 1.310e-14
## as_factor(cowcode)666        -0.99110185  0.37269824 -2.6593 0.0078312
## as_factor(cowcode)670        -3.02601360  0.45470666 -6.6549 2.836e-11
## as_factor(cowcode)679        -2.65474505  0.36791325 -7.2157 5.367e-13
```

```
## as_factor(cowcode)690         -1.94251283  0.44274298 -4.3875 1.147e-05
## as_factor(cowcode)694         -1.64762151  0.51544130 -3.1965 0.0013909
## as_factor(cowcode)698         -2.36790989  0.43594516 -5.4317 5.583e-08
## as_factor(cowcode)700         -2.20141921  0.36008126 -6.1137 9.736e-10
## as_factor(cowcode)701         -1.52070627  0.59015023 -2.5768 0.0099716
## as_factor(cowcode)702         -1.06102962  0.25282619 -4.1967 2.709e-05
## as_factor(cowcode)703         -0.85184361  0.25238078 -3.3752 0.0007375
## as_factor(cowcode)704         -0.61807113  0.41198759 -1.5002 0.1335580
## as_factor(cowcode)705         -0.27628978  0.30680122 -0.9005 0.3678277
## as_factor(cowcode)710         -0.45446138  0.27542714 -1.6500 0.0989380
## as_factor(cowcode)712         -0.46548852  0.25690139 -1.8119 0.0699963
## as_factor(cowcode)732         -1.32631696  0.34790572 -3.8123 0.0001377
## as_factor(cowcode)750         -1.69525302  0.28699314 -5.9069 3.485e-09
## as_factor(cowcode)760         -0.32587319  0.25235829 -1.2913 0.1965956
## as_factor(cowcode)770         -1.93075049  0.32463556 -5.9474 2.724e-09
## as_factor(cowcode)771         -1.55113222  0.30171909 -5.1410 2.733e-07
## as_factor(cowcode)775         -0.64376341  0.33557209 -1.9184 0.0550596
## as_factor(cowcode)780         -1.73755314  0.27656425 -6.2826 3.329e-10
## as_factor(cowcode)781         -0.96468056  0.31142822 -3.0976 0.0019509
## as_factor(cowcode)790          0.80325309  0.26146251  3.0722 0.0021252
## as_factor(cowcode)800         -0.36746124  0.26693288 -1.3766 0.1686343
## as_factor(cowcode)811          0.74955292  0.26174922  2.8636 0.0041882
## as_factor(cowcode)812          0.40907889  0.26632604  1.5360 0.1245365
## as_factor(cowcode)820         -1.18741302  0.31151628 -3.8117 0.0001380
## as_factor(cowcode)830         -1.60478110  0.41428818 -3.8736 0.0001072
## as_factor(cowcode)840         -0.70497321  0.25301857 -2.7863 0.0053322
## as_factor(cowcode)850         -0.65217010  0.27091241 -2.4073 0.0160705
## as_factor(cowcode)910          0.49161907  0.28118844  1.7484 0.0804014
## as_factor(cowcode)935         -0.02301878  0.27172678 -0.0847 0.9324896
## as_factor(cowcode)940          0.18645958  0.25255997  0.7383 0.4603453
## as_factor(cowcode)950         -1.37757452  0.28332285 -4.8622 1.161e-06
##
## Rho: 0.10614, LR test value: 6.711, p-value: 0.0095822
## Asymptotic standard error: 0.037627
##     z-value: 2.8209, p-value: 0.0047884
## Wald statistic: 7.9577, p-value: 0.0047884
##
## Log likelihood: -3371.868 for lag model
## ML residual variance (sigma squared): 0.56917, (sigma: 0.75444)
## Number of observations: 2964
## Number of parameters estimated: 157
## AIC: 7057.7, (AIC for lm: 7062.4)
## LM test for residual autocorrelation
## test value: 4.7177, p-value: 0.029853
```

```
# Table 2: FLFP ~ REER  + Country Effects    | REER @ 80% HDI  ----------

# What's the highest density 80% of the distribution?
# You may need to install the package `bayestestR` package for the code below

df_tscs %>%
  filter(oecd == 0 & country != "Zimbabwe") |>
  summarize(hdi = bayestestR::hdi(log_wdi_overvalued, ci = .8))
```

```
## # A tibble: 1 x 1
```

```
##   hdi$CI $CI_low $CI_high
##    <dbl>   <dbl>    <dbl>
## 1    0.8  -0.484    0.644
```

```
m_flfp_strict <- plm(
  dflfp ~ flfplag + dflfplag +
    log_wdi_overvaluedlag + dlog_wdi_overvalued +
    log_gdpcaplag + dlog_gdpcap +
    sqlog_gdpcaplag + dsqlog_gdpcap +
    log_oilgascaplag + dlog_oilgascap - country,
  data = subset(
    df_tscs,
    oecd == 0 &
      country != "Zimbabwe" &
      log_wdi_overvalued >= -.48 &
      log_wdi_overvalued <= .64
  ),
  model = "within",
  effect = "individual",
  index = c("cowcode", "year")
)

# View the results
show_pcse(m_flfp_strict)
```

```
## Oneway (individual) effect Within Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovBK(x, type = "HC1", cluster = "time")
##
## Call:
## plm(formula = dflfp ~ flfplag + dflfplag + log_wdi_overvaluedlag +
##     dlog_wdi_overvalued + log_gdpcaplag + dlog_gdpcap + sqlog_gdpcaplag +
##     dsqlog_gdpcap + log_oilgascaplag + dlog_oilgascap - country,
##     data = subset(df_tscs, oecd == 0 & country != "Zimbabwe" &
##         log_wdi_overvalued >= -0.48 & log_wdi_overvalued <= 0.64),
##     effect = "individual", model = "within", index = c("cowcode",
##         "year"))
##
## Unbalanced Panel: n = 137, T = 1-24, N = 2644
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -6.5083176 -0.2144877  0.0045127  0.2167290  6.4112919
##
## Coefficients:
##                          Estimate Std. Error t-value  Pr(>|t|)
## flfplag                 -0.049639   0.011047 -4.4935 7.326e-06 ***
## dflfplag                 0.145741   0.060343  2.4152 0.0157970 *
## log_wdi_overvaluedlag -0.200548   0.078211 -2.5642 0.0103996 *
## dlog_wdi_overvalued   -0.335329   0.121962 -2.7494 0.0060126 **
## log_gdpcaplag         -1.345529   0.397664 -3.3836 0.0007265 ***
## dlog_gdpcap           -4.929946   1.520713 -3.2419 0.0012031 **
## sqlog_gdpcaplag        0.096008   0.025962  3.6980 0.0002220 ***
## dsqlog_gdpcap          0.354865   0.097365  3.6447 0.0002732 ***
## log_oilgascaplag      -0.022503   0.024892 -0.9040 0.3660617
```

```
## dlog_oilgascap         0.036253   0.048202  0.7521 0.4520519
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    1613.9
## Residual Sum of Squares: 1517.7
## R-Squared:      0.05962
## Adj. R-Squared: 0.0046364
## F-statistic: 6.328 on 10 and 23 DF, p-value: 0.00012843
```

```
# Table 2: FLFP ~ REER  + Extended Controls  | REER @ 80% HDI  ----------

# First, re-create the relevant data
df <- df_tscs %>%
  filter(
    oecd == 0 &
      country_space != "Zimbabwe (Rhodesia)" &
      country_space != "Tonga" &
      log_wdi_overvalued >= -.48 &
      log_wdi_overvalued <= .64
  )

# Estimate the model
pre <- lm(
  dflfp ~ flfplag + dflfplag +
    log_wdi_overvaluedlag + dlog_wdi_overvalued +
    log_gdpcaplag + dlog_gdpcap +
    sqlog_gdpcaplag + dsqlog_gdpcap +
    log_oilgascaplag + dlog_oilgascap +
    as_factor(irr_rate_regimelag) +
    wdi_fertilitylag + dwdi_fertility +
    vdem_elec_demlag + dvdem_elec_dem +
    as_factor(cowcode) -
    cowcode - country - country_space - year,
  data = df
)

# Create the usable data frame for the SLX model
df2 <- pre$model %>% data.frame() %>% clean_names()

# Make the spatial weights where k = 10
W <- make_ntspmat(pre, ci = country_space, y = year, k = 10)
```

```
## [1] 1992
##  [1] 42  51  52  70  90  91  93  94  95  100 101 110 135 140 145 150 155 165 290 310 355 411 432 434
## [38] 517 540 551 553 560 565 570 571 572 580 590 600 615 616 640 660 663 670 710 712 750 760 770 771
## [1] 1993
##  [1] 42  51  52  90  91  93  94  95  100 101 110 135 140 145 150 155 165 290 310 339 355 360 411 432
## [38] 530 540 551 553 560 565 570 571 572 580 590 600 615 616 640 651 660 663 670 710 750 760 770 771
## [1] All of your Countries are Matched.
## [1] 1994
##  [1] 42  51  52  90  91  93  94  95  100 101 110 135 140 145 150 290 310 316 339 355 360 369 411 432
## [38] 490 500 501 510 516 530 551 553 560 570 571 572 580 581 590 600 615 616 640 651 660 663 670 710
## [1] All of your Countries are Matched.
## [1] 1995
```

```
##  [1] 42  51  52  70  90  91  93  94  95  100 101 110 115 135 140 145 150 290 310 316 339 343 355 360
## [38] 471 475 481 483 484 490 500 501 510 516 530 540 551 553 560 565 570 571 572 580 581 590 600 615
## [75] 770 771 780 790 800 811 812 820 840 850 910
## [1] All of your Countries are Matched.
## [1] 1996
##  [1] 42  51  52  70  90  91  93  94  95  100 101 110 115 135 145 150 290 310 316 339 343 344 355 359
## [38] 461 471 481 483 484 490 500 501 510 516 517 530 540 551 553 560 565 570 571 572 580 581 590 600
## [75] 705 710 712 750 760 770 771 780 781 790 800 811 812 820 840 850 910 950
## [1] All of your Countries are Matched.
## [1] 1997
##  [1] 41  42  51  52  70  90  91  93  94  95  100 101 110 115 135 145 150 290 310 316 339 343 344 355
## [38] 437 438 439 451 452 461 471 481 482 483 484 490 500 501 510 516 530 540 551 553 560 565 570 571
## [75] 670 690 702 703 704 705 710 750 760 770 771 780 781 790 800 811 812 820 840 850 910
## [1] All of your Countries are Matched.
## [1] 1998
##  [1] 41  42  52  70  90  91  93  94  95  100 101 110 115 135 145 150 290 310 316 339 343 346 355 359
## [38] 438 439 451 452 461 471 481 483 484 490 500 501 510 516 530 531 540 551 553 560 565 570 571 572
## [75] 702 703 704 705 710 732 750 760 770 771 780 781 790 800 811 812 820 840 910 950
## [1] All of your Countries are Matched.
## [1] 1999
##  [1] 41  42  51  52  70  90  91  93  94  95  100 101 110 115 135 140 145 150 155 290 310 316 339 343
## [38] 404 432 433 434 436 437 438 439 451 452 461 471 481 482 483 484 500 501 510 516 517 530 551 553
## [75] 640 651 660 663 670 690 702 703 704 710 732 750 760 771 780 781 790 800 811 820 830 840 910 950
## [1] All of your Countries are Matched.
## [1] 2000
##  [1] 41  42  51  52  70  90  91  93  94  95  100 101 110 115 135 140 145 150 155 290 310 316 339 343
## [38] 433 434 436 437 438 439 451 461 471 481 482 483 484 500 501 510 516 517 530 540 551 553 560 565
## [75] 651 660 663 670 690 703 704 710 732 750 760 770 771 780 781 790 800 811 820 830 840 910 950
## [1] All of your Countries are Matched.
## [1] 2001
##  [1] 41  42  51  52  70  90  91  92  93  94  95  100 101 110 135 140 145 150 155 160 165 290 310 316
## [38] 420 432 433 434 436 437 438 439 451 461 471 481 482 483 484 500 501 510 516 517 530 540 551 553
## [75] 630 640 651 660 663 670 690 694 703 710 732 750 760 771 781 790 800 811 820 830 840 910 950
## [1] All of your Countries are Matched.
## [1] 2002
##  [1] 41  42  51  52  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290 310
## [38] 402 404 420 432 433 434 436 437 438 439 451 461 471 481 482 483 484 490 500 501 510 516 517 530
## [75] 616 625 640 660 663 666 670 690 694 710 732 750 760 771 781 790 800 811 820 830 840 910 950
## [1] All of your Countries are Matched.
## [1] 2003
##   [1] 41  42  51  52  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165 290
##  [38] 372 402 404 420 432 433 434 436 437 438 439 451 461 471 481 482 483 484 490 500 501 510 516 517
##  [75] 600 616 625 640 660 663 666 670 690 694 700 703 710 732 750 760 771 781 790 800 811 820 830 840
## [1] All of your Countries are Matched.
## [1] 2004
##   [1] 41  42  51  52  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165 290
##  [38] 372 402 404 420 432 433 434 436 437 438 439 451 461 471 481 482 483 484 490 500 501 510 516 517
##  [75] 590 600 615 616 620 625 640 660 663 666 670 690 694 698 700 702 703 710 732 750 760 771 781 790
## [1] All of your Countries are Matched.
## [1] 2005
##   [1] 41  42  51  52  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165 290
##  [38] 371 372 402 404 420 432 433 434 436 437 438 439 451 461 471 481 482 483 484 490 500 501 510 516
##  [75] 590 600 615 616 620 625 640 660 663 666 670 690 694 698 700 702 703 710 732 750 760 771 781 790
## [1] All of your Countries are Matched.
```

```
## [1] 2006
##   [1] 41  42  51  52  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165 290
##  [38] 371 372 402 404 420 432 433 434 436 437 438 439 451 452 461 475 481 482 483 484 490 500 501 510
##  [75] 581 590 600 615 616 620 625 640 660 663 666 670 690 694 698 700 702 703 705 710 732 750 760 771
## [1] All of your Countries are Matched.
## [1] 2007
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 369 370 371 372 373 402 404 411 420 432 433 434 436 437 438 439 451 452 461 481 482 483 484 490
##  [75] 571 572 580 581 590 600 615 616 620 625 640 660 663 666 670 690 694 698 700 702 703 705 710 731
## [1] All of your Countries are Matched.
## [1] 2008
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 369 370 371 372 373 402 404 411 420 432 433 434 436 437 438 439 451 452 461 471 475 481 482 483
##  [75] 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 660 663 670 679 690 694 698 700
## [112] 820 830 840 850 910 940 950
## [1] All of your Countries are Matched.
## [1] 2009
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 369 370 371 372 373 402 404 411 420 432 433 434 436 437 438 439 451 452 461 471 481 482 483
##  [75] 560 565 570 571 572 580 581 590 600 616 620 625 630 640 660 663 666 670 690 694 698 700 702 703
## [112] 940 950
## [1] All of your Countries are Matched.
## [1] 2010
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 369 370 371 372 373 402 404 411 420 432 433 434 436 437 438 439 451 461 471 475 481 482 483
##  [75] 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 660 663 666 670 690 694 698 700
## [112] 811 820 830 840 850 910 940 950
## [1] All of your Countries are Matched.
## [1] 2011
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 369 370 371 372 373 402 404 411 420 432 433 434 436 437 438 439 451 461 471 475 481 482 483
##  [75] 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 660 663 666 670 679 690 694 698
## [112] 820 830 840 850 910 940 950
## [1] All of your Countries are Matched.
## [1] 2012
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 369 370 371 372 373 402 404 411 420 432 433 434 436 437 438 439 451 461 471 475 481 482 483
##  [75] 565 570 571 572 580 581 590 600 615 616 625 630 640 645 660 663 666 670 679 690 694 698 700 701
## [112] 820 830 840 850 910 940 950
## [1] All of your Countries are Matched.
## [1] 2013
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 369 370 371 372 373 402 404 411 420 432 433 434 436 437 438 439 451 461 471 475 481 482 483
##  [75] 565 570 571 572 580 581 590 600 615 616 625 630 640 645 660 663 666 670 679 690 694 698 700 701
## [112] 830 840 850 910 940 950
## [1] All of your Countries are Matched.
## [1] 2014
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 140 145 150 155 160 165 290
##  [38] 370 371 372 373 402 404 411 432 433 434 437 438 439 451 452 461 471 475 481 482 483 484 490 500
##  [75] 572 580 581 590 600 615 616 625 640 645 660 663 666 670 679 690 694 698 700 702 703 705 710 711
## [112] 940 950
## [1] All of your Countries are Matched.
## [1] 2015
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 110 115 130 140 145 150 155 160 165 290 310
```

```
##  [38] 372 402 404 411 420 432 433 434 437 438 439 451 452 461 471 475 481 482 483 490 500 501 510 510
##  [75] 590 600 616 625 640 645 651 660 663 666 670 679 690 694 698 700 702 703 705 710 712 732 760 770
## [1] All of your Countries are Matched.
```

```
# Estimate the SAR model (may take a couple of minutes)
m_flfp_strict_sar <- lagsarlm(
  dflfp ~ flfplag + dflfplag +
    log_wdi_overvaluedlag + dlog_wdi_overvalued +
    log_gdpcaplag + dlog_gdpcap +
    sqlog_gdpcaplag + dsqlog_gdpcap +
    log_oilgascaplag + dlog_oilgascap +
    as_factor_irr_rate_regimelag +
    wdi_fertilitylag + dwdi_fertility +
    vdem_elec_demlag + dvdem_elec_dem +
    as_factor(cowcode),
  listw = mat2listw(W[[2]], style = "W"),
  Durbin = FALSE,
  data = df2
)

# View the results
summary(m_flfp_strict_sar)
```

```
##
## Call:lagsarlm(formula = dflfp ~ flfplag + dflfplag + log_wdi_overvaluedlag +     dlog_wdi_overvalued
##     dsqlog_gdpcap + log_oilgascaplag + dlog_oilgascap + as_factor_irr_rate_regimelag +
##     wdi_fertilitylag + dwdi_fertility + vdem_elec_demlag + dvdem_elec_dem +     as_factor(cowcode), d
##     style = "W"), Durbin = FALSE)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -6.682638 -0.231587  0.010062  0.222278  6.466879
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##                                 Estimate Std. Error z value  Pr(>|z|)
## (Intercept)                   10.6905760  2.6694452  4.0048 6.207e-05
## flfplag                       -0.0544290  0.0057990 -9.3859 < 2.2e-16
## dflfplag                       0.1195821  0.0205009  5.8330 5.444e-09
## log_wdi_overvaluedlag         -0.1788109  0.1006797 -1.7760 0.0757268
## dlog_wdi_overvalued           -0.3594682  0.1390536 -2.5851 0.0097349
## log_gdpcaplag                 -1.8503671  0.6335471 -2.9206 0.0034931
## dlog_gdpcap                   -6.2502628  2.0197284 -3.0946 0.0019707
## sqlog_gdpcaplag                0.1290145  0.0393876  3.2755 0.0010547
## dsqlog_gdpcap                  0.4541280  0.1287842  3.5263 0.0004215
## log_oilgascaplag              -0.0316387  0.0350028 -0.9039 0.3660532
## dlog_oilgascap                 0.0202080  0.0570030  0.3545 0.7229580
## as_factor_irr_rate_regimelag2 -0.1079475  0.0773655 -1.3953 0.1629280
## as_factor_irr_rate_regimelag3 -0.0190448  0.0872648 -0.2182 0.8272406
## as_factor_irr_rate_regimelag4  0.0367407  0.1872673  0.1962 0.8444586
## as_factor_irr_rate_regimelag5  0.0390268  0.0995233  0.3921 0.6949564
## as_factor_irr_rate_regimelag6  0.0046425  0.2463093  0.0188 0.9849622
## wdi_fertilitylag              -0.0896991  0.0509670 -1.7599 0.0784170
## dwdi_fertility                -0.1175549  0.4068832 -0.2889 0.7726459
## vdem_elec_demlag              -0.5266213  0.2149348 -2.4501 0.0142799
```

```
## dvdem_elec_dem                0.0589596  0.3715516  0.1587 0.8739173
## as_factor(cowcode)42         -0.6191539  0.2987221 -2.0727 0.0382026
## as_factor(cowcode)51         -0.4869675  0.3014670 -1.6153 0.1062401
## as_factor(cowcode)52         -1.0449498  0.3774429 -2.7685 0.0056315
## as_factor(cowcode)53         -0.8928601  0.4175293 -2.1384 0.0324813
## as_factor(cowcode)70         -1.2021897  0.3655818 -3.2884 0.0010075
## as_factor(cowcode)90         -1.0326254  0.2940620 -3.5116 0.0004454
## as_factor(cowcode)91         -0.7801452  0.2761411 -2.8252 0.0047256
## as_factor(cowcode)92         -0.9970663  0.3196873 -3.1189 0.0018154
## as_factor(cowcode)93         -0.6844241  0.2713889 -2.5219 0.0116713
## as_factor(cowcode)94         -0.8693371  0.3421710 -2.5407 0.0110646
## as_factor(cowcode)95         -0.8598422  0.3308144 -2.5992 0.0093450
## as_factor(cowcode)100        -0.2154851  0.3295543 -0.6539 0.5131967
## as_factor(cowcode)101        -1.0144893  0.3783464 -2.6814 0.0073320
## as_factor(cowcode)110        -1.4509597  0.2999328 -4.8376 1.314e-06
## as_factor(cowcode)115        -1.1897951  0.3683621 -3.2300 0.0012381
## as_factor(cowcode)130        -0.6876324  0.3640939 -1.8886 0.0589437
## as_factor(cowcode)135         0.7988546  0.2964384  2.6948 0.0070422
## as_factor(cowcode)140        -0.6789083  0.3521805 -1.9277 0.0538889
## as_factor(cowcode)145        -0.0239010  0.2897312 -0.0825 0.9342542
## as_factor(cowcode)150        -0.3939104  0.2797931 -1.4079 0.1591716
## as_factor(cowcode)155        -0.9433030  0.3746660 -2.5177 0.0118118
## as_factor(cowcode)160        -1.1286919  0.3618564 -3.1192 0.0018136
## as_factor(cowcode)165        -0.7805080  0.3645458 -2.1410 0.0322706
## as_factor(cowcode)290        -1.3952744  0.3444255 -4.0510 5.100e-05
## as_factor(cowcode)310        -1.6089630  0.3586502 -4.4862 7.252e-06
## as_factor(cowcode)316        -1.4914121  0.3808932 -3.9156 9.019e-05
## as_factor(cowcode)339        -1.1482010  0.2893333 -3.9684 7.235e-05
## as_factor(cowcode)341        -1.7328904  0.3926465 -4.4134 1.018e-05
## as_factor(cowcode)343        -1.4439925  0.3153859 -4.5785 4.683e-06
## as_factor(cowcode)344        -1.5234549  0.3715403 -4.1004 4.125e-05
## as_factor(cowcode)345        -1.4040741  0.3748926 -3.7453 0.0001802
## as_factor(cowcode)346        -2.0576258  0.3447729 -5.9681 2.401e-09
## as_factor(cowcode)355        -1.4157240  0.3097649 -4.5703 4.870e-06
## as_factor(cowcode)359        -1.4546747  0.2895545 -5.0238 5.065e-07
## as_factor(cowcode)360        -1.5617316  0.3240466 -4.8195 1.439e-06
## as_factor(cowcode)365        -0.8213514  0.3617978 -2.2702 0.0231958
## as_factor(cowcode)367        -1.1901050  0.3862877 -3.0809 0.0020639
## as_factor(cowcode)369        -1.2416937  0.3201976 -3.8779 0.0001054
## as_factor(cowcode)370        -0.6970006  0.3111760 -2.2399 0.0250979
## as_factor(cowcode)371        -0.9670728  0.2838780 -3.4066 0.0006577
## as_factor(cowcode)372        -0.5384599  0.2806350 -1.9187 0.0550198
## as_factor(cowcode)373        -0.5169387  0.3703544 -1.3958 0.1627763
## as_factor(cowcode)402        -0.7135060  0.3227697 -2.2106 0.0270654
## as_factor(cowcode)404         0.0815843  0.2795727  0.2918 0.7704260
## as_factor(cowcode)411        -0.7596041  0.3971813 -1.9125 0.0558137
## as_factor(cowcode)420        -0.5147624  0.2967887 -1.7344 0.0828399
## as_factor(cowcode)432        -0.1218131  0.3105484 -0.3923 0.6948723
## as_factor(cowcode)433        -0.7703797  0.3051959 -2.5242 0.0115957
## as_factor(cowcode)434         0.6891318  0.2849085  2.4188 0.0155725
## as_factor(cowcode)436         0.3763816  0.3285471  1.1456 0.2519632
## as_factor(cowcode)437        -0.6147217  0.3014523 -2.0392 0.0414300
## as_factor(cowcode)438         0.0653666  0.2727440  0.2397 0.8105915
## as_factor(cowcode)439        -0.3676080  0.2934122 -1.2529 0.2102522
```

```
## as_factor(cowcode)451        -0.2926326  0.2686906 -1.0891 0.2761070
## as_factor(cowcode)452         0.7514473  0.3060782  2.4551 0.0140852
## as_factor(cowcode)461         0.6650934  0.2893096  2.2989 0.0215107
## as_factor(cowcode)471         0.6005182  0.3258029  1.8432 0.0653006
## as_factor(cowcode)475        -0.2428055  0.3955991 -0.6138 0.5393696
## as_factor(cowcode)481        -1.4886610  0.4039199 -3.6855 0.0002282
## as_factor(cowcode)482        -0.1012396  0.2839195 -0.3566 0.7214072
## as_factor(cowcode)483         0.2499287  0.3185933  0.7845 0.4327611
## as_factor(cowcode)484         0.3894923  0.3672152  1.0607 0.2888422
## as_factor(cowcode)490         0.4183316  0.3064771  1.3650 0.1722629
## as_factor(cowcode)500         0.2363796  0.2775411  0.8517 0.3943849
## as_factor(cowcode)501        -0.0625408  0.2615384 -0.2391 0.8110075
## as_factor(cowcode)510         0.9995011  0.2980729  3.3532 0.0007988
## as_factor(cowcode)516         0.5552461  0.3389980  1.6379 0.1014418
## as_factor(cowcode)517         1.1543777  0.3140904  3.6753 0.0002376
## as_factor(cowcode)530         0.7437041  0.3100835  2.3984 0.0164669
## as_factor(cowcode)531         0.4036530  0.3981128  1.0139 0.3106228
## as_factor(cowcode)540         0.8763082  0.4031227  2.1738 0.0297201
## as_factor(cowcode)541         1.0244954  0.3403299  3.0103 0.0026099
## as_factor(cowcode)551         0.5164100  0.3003209  1.7195 0.0855184
## as_factor(cowcode)553         0.4549855  0.2791501  1.6299 0.1031236
## as_factor(cowcode)560        -1.0956191  0.3483122 -3.1455 0.0016580
## as_factor(cowcode)565        -0.4360881  0.3168655 -1.3763 0.1687423
## as_factor(cowcode)570        -0.2219419  0.2591545 -0.8564 0.3917723
## as_factor(cowcode)571        -0.0807717  0.3063388 -0.2637 0.7920359
## as_factor(cowcode)572        -1.4805643  0.3248119 -4.5582 5.159e-06
## as_factor(cowcode)580         1.0386112  0.2971180  3.4956 0.0004730
## as_factor(cowcode)581        -1.3845219  0.2996528 -4.6204 3.830e-06
## as_factor(cowcode)590        -1.2140660  0.3342347 -3.6324 0.0002808
## as_factor(cowcode)600        -2.2648200  0.3368724 -6.7231 1.779e-11
## as_factor(cowcode)615        -2.7226552  0.4283756 -6.3558 2.074e-10
## as_factor(cowcode)616        -2.2252326  0.3675949 -6.0535 1.417e-09
## as_factor(cowcode)620        -2.3818129  0.4772281 -4.9909 6.009e-07
## as_factor(cowcode)625        -2.1928769  0.3624892 -6.0495 1.453e-09
## as_factor(cowcode)630        -2.9134818  0.4713506 -6.1811 6.364e-10
## as_factor(cowcode)640        -2.5677283  0.3896222 -6.5903 4.389e-11
## as_factor(cowcode)645        -2.2800561  0.5022545 -4.5396 5.635e-06
## as_factor(cowcode)651        -2.5535470  0.4194397 -6.0880 1.143e-09
## as_factor(cowcode)660        -2.7362341  0.4031104 -6.7878 1.139e-11
## as_factor(cowcode)663        -2.8676784  0.3911807 -7.3308 2.287e-13
## as_factor(cowcode)666        -1.1117311  0.4516181 -2.4617 0.0138295
## as_factor(cowcode)670        -3.2503923  0.5053775 -6.4316 1.263e-10
## as_factor(cowcode)679        -2.9947112  0.4687921 -6.3881 1.679e-10
## as_factor(cowcode)690        -2.1532356  0.5082942 -4.2362 2.273e-05
## as_factor(cowcode)694        -1.8713170  0.6016474 -3.1103 0.0018688
## as_factor(cowcode)698        -2.4697605  0.4903290 -5.0369 4.730e-07
## as_factor(cowcode)700        -2.2700278  0.3883847 -5.8448 5.072e-09
## as_factor(cowcode)702        -1.1304727  0.2857444 -3.9562 7.614e-05
## as_factor(cowcode)703        -0.9196860  0.2684973 -3.4253 0.0006141
## as_factor(cowcode)704        -0.6741205  0.4377240 -1.5401 0.1235461
## as_factor(cowcode)705        -0.2438983  0.3662775 -0.6659 0.5054853
## as_factor(cowcode)710        -0.4953177  0.2977747 -1.6634 0.0962329
## as_factor(cowcode)712        -0.4269923  0.3343343 -1.2771 0.2015522
## as_factor(cowcode)732        -1.6576362  0.4186414 -3.9596 7.509e-05
```

```
## as_factor(cowcode)750         -1.6591440  0.3146195 -5.2735 1.339e-07
## as_factor(cowcode)760         -0.0585698  0.2698668 -0.2170 0.8281832
## as_factor(cowcode)770         -2.2323258  0.4119261 -5.4192 5.985e-08
## as_factor(cowcode)771         -1.6788102  0.3242958 -5.1768 2.257e-07
## as_factor(cowcode)775         -1.8525714  1.0075693 -1.8387 0.0659661
## as_factor(cowcode)780         -1.8448503  0.3261978 -5.6556 1.553e-08
## as_factor(cowcode)781         -1.0147122  0.3396687 -2.9874 0.0028140
## as_factor(cowcode)790          0.8280684  0.2791555  2.9663 0.0030137
## as_factor(cowcode)800         -0.4426517  0.2866619 -1.5442 0.1225497
## as_factor(cowcode)811          0.7741191  0.2753164  2.8117 0.0049274
## as_factor(cowcode)812          0.6220296  0.3342342  1.8611 0.0627358
## as_factor(cowcode)820         -1.3011941  0.3399502 -3.8276 0.0001294
## as_factor(cowcode)830         -1.8522314  0.4954021 -3.7388 0.0001849
## as_factor(cowcode)840         -0.7223427  0.2685576 -2.6897 0.0071514
## as_factor(cowcode)850         -0.7632332  0.3314810 -2.3025 0.0213073
## as_factor(cowcode)910          0.5833528  0.3097524  1.8833 0.0596614
## as_factor(cowcode)935          0.0099094  0.8193749  0.0121 0.9903507
## as_factor(cowcode)940          0.1458230  0.3086770  0.4724 0.6366320
## as_factor(cowcode)950         -1.4256205  0.3136079 -4.5459 5.471e-06
##
## Rho: 0.063595, LR test value: 1.909, p-value: 0.16707
## Asymptotic standard error: 0.042822
##     z-value: 1.4851, p-value: 0.13752
## Wald statistic: 2.2055, p-value: 0.13752
##
## Log likelihood: -2842.359 for lag model
## ML residual variance (sigma squared): 0.60726, (sigma: 0.77927)
## Number of observations: 2430
## Number of parameters estimated: 148
## AIC: 5980.7, (AIC for lm: 5980.6)
## LM test for residual autocorrelation
## test value: 8.2403, p-value: 0.0040969
```

```r
# Table 2: Ratio ~ REER + Country Effects ------------------------------------------------

m_ratio_base <-
  plm(
    dflfp_mlfp ~ flfp_mlfplag + dflfp_mlfplag +
      log_wdi_overvaluedlag + dlog_wdi_overvalued +
      log_gdpcaplag + dlog_gdpcap + sqlog_gdpcaplag +
      dsqlog_gdpcap + log_oilgascaplag + dlog_oilgascap,
    data = subset(df_tscs, oecd == 0 & country != "Zimbabwe"),
    model = "within",
    effect = "individual",
    index = c("cowcode", "year")
  )

# View results with PCSEs
show_pcse(m_ratio_base)
```

```
## Oneway (individual) effect Within Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovBK(x, type = "HC1", cluster = "time")
##
## Call:
```

```
## plm(formula = dflfp_mlfp ~ flfp_mlfplag + dflfp_mlfplag + log_wdi_overvaluedlag +
##     dlog_wdi_overvalued + log_gdpcaplag + dlog_gdpcap + sqlog_gdpcaplag +
##     dsqlog_gdpcap + log_oilgascaplag + dlog_oilgascap, data = subset(df_tscs,
##     oecd == 0 & country != "Zimbabwe"), effect = "individual",
##     model = "within", index = c("cowcode", "year"))
##
## Unbalanced Panel: n = 147, T = 5-24, N = 3245
##
## Residuals:
##        Min.     1st Qu.      Median     3rd Qu.        Max.
## -14.9208864  -0.2983591   0.0085417   0.3043556   8.8496390
##
## Coefficients:
##                          Estimate Std. Error t-value  Pr(>|t|)
## flfp_mlfplag          -0.0487939  0.0113452 -4.3008 1.754e-05 ***
## dflfp_mlfplag          0.0938361  0.0555558  1.6890 0.0913124 .
## log_wdi_overvaluedlag -0.1930010  0.0492124 -3.9218 8.980e-05 ***
## dlog_wdi_overvalued   -0.2493491  0.0676003 -3.6886 0.0002294 ***
## log_gdpcaplag         -0.9805196  0.4280793 -2.2905 0.0220589 *
## dlog_gdpcap            2.9950976  1.2866153  2.3279 0.0199823 *
## sqlog_gdpcaplag        0.0715368  0.0279108  2.5630 0.0104227 *
## dsqlog_gdpcap         -0.0615037  0.0914226 -0.6727 0.5011626
## log_oilgascaplag       0.0038755  0.0194279  0.1995 0.8419005
## dlog_oilgascap         0.0548229  0.0447180  1.2260 0.2203040
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    2784.6
## Residual Sum of Squares: 2623.8
## R-Squared:      0.057722
## Adj. R-Squared: 0.01012
## F-statistic: 12.3381 on 10 and 23 DF, p-value: 4.8393e-07
```

```
# Table 2: Ratio ~ REER + Extended Controls ------------------------

# Create the usable dataset
df <- df_tscs %>%
  filter(oecd == 0 &
           country_space != "Zimbabwe (Rhodesia)" &
           country_space != "Tonga")

pre <- lm(
  dflfp_mlfp ~ flfp_mlfplag + dflfp_mlfplag +
    log_wdi_overvaluedlag + dlog_wdi_overvalued +
    log_gdpcaplag + dlog_gdpcap +
    sqlog_gdpcaplag + dsqlog_gdpcap +
    log_oilgascaplag + dlog_oilgascap +
    as_factor(irr_rate_regimelag) +
    wdi_fertilitylag + dwdi_fertility +
    vdem_elec_demlag + dvdem_elec_dem +
    as_factor(cowcode) - cowcode - year - country_space,
  data = df
)
```

```r
df2 <- pre$model %>% data.frame() %>% clean_names()

# Make the spatial weights where k = 10
W <- make_ntspmat(pre, ci = country_space, y = year, k = 10)
```

```
## [1] 1992
##  [1] 42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290 310
## [38] 438 439 450 451 452 461 471 475 481 482 483 484 490 500 501 510 516 517 530 540 541 551 553 560
## [75] 663 666 670 679 710 712 732 750 760 770 771 780 790 800 812 820 830 840 850 910 935 940 950
## [1] 1993
##   [1] 42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290 310
##  [38] 436 437 438 439 450 451 452 461 471 475 481 482 483 484 490 500 501 510 516 517 530 540 541 551
##  [75] 651 660 663 666 670 679 702 710 712 732 750 760 770 771 780 790 800 812 820 830 840 850 910 935
## [1] All of your Countries are Matched.
## [1] 1994
##   [1] 42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290 310
##  [38] 433 434 435 436 437 438 439 450 451 452 461 471 475 481 482 483 484 490 500 501 510 516 517 530
##  [75] 615 616 630 640 651 660 663 666 670 679 702 710 712 732 750 760 770 771 780 790 800 811 812 820
## [1] All of your Countries are Matched.
## [1] 1995
##   [1] 42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290 310
##  [38] 411 420 432 433 434 435 436 437 438 439 450 451 452 461 471 475 481 482 483 484 490 500 501 510
##  [75] 581 590 600 615 616 630 640 651 660 663 666 670 679 701 702 703 705 710 712 732 750 760 770 771
## [1] All of your Countries are Matched.
## [1] 1996
##   [1] 42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290 310
##  [38] 369 370 371 373 402 404 411 420 432 433 434 435 436 437 438 439 450 451 452 461 471 475 481 482
##  [75] 560 565 570 571 572 580 581 590 600 615 616 630 640 651 660 663 666 670 679 690 701 702 703 704
## [112] 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 1997
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290
##  [38] 367 368 369 370 371 372 373 402 404 411 420 432 433 434 435 436 437 438 439 450 451 452 461 471
##  [75] 541 551 553 560 565 570 571 572 580 581 590 600 615 616 625 630 640 651 660 663 666 670 679 690
## [112] 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 1998
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290
##  [38] 366 367 368 369 370 371 372 373 402 404 411 420 432 433 434 435 436 437 438 439 450 451 452 461
##  [75] 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 625 630 640 651 660 663 666 670
## [112] 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 1999
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290
##  [38] 366 367 368 369 370 371 372 373 402 404 411 420 432 433 434 435 436 437 438 439 450 451 452 461
##  [75] 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 625 630 640 651 660 663 666 670
## [112] 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2000
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290
##  [38] 366 367 368 369 370 371 372 373 402 404 420 432 433 434 435 436 437 438 439 450 451 452 461 471
##  [75] 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 651 660 663 666 670
## [112] 771 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
```

```
## [1] 2001
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290
##  [38] 366 367 368 369 370 371 372 373 402 404 420 432 433 434 435 436 437 438 439 450 451 452 461 471
##  [75] 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 651 660 663 666 670
## [112] 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2002
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290
##  [38] 366 367 368 369 370 371 372 373 402 403 404 420 432 433 434 435 436 437 438 439 450 451 452 461
##  [75] 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 651 660 663 660
## [112] 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2003
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 365 366 367 368 369 370 371 372 373 402 403 404 420 432 433 434 435 436 437 438 439 450 451 451
##  [75] 530 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 651 660 663
## [112] 750 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2004
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 365 366 367 368 369 370 371 372 373 402 403 404 420 432 433 434 436 437 438 439 450 451 452 461
##  [75] 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 651 660 663 660
## [112] 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2005
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 365 366 367 368 369 370 371 372 373 402 403 404 420 432 433 434 436 437 438 439 450 451 452 461
##  [75] 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 651 660 663
## [112] 750 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2006
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 365 366 367 368 369 370 371 372 373 402 403 404 420 432 433 434 435 436 437 438 439 450 451 451
##  [75] 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 651 660 663
## [112] 750 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2007
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 360 365 366 367 368 369 370 371 372 373 402 403 404 411 420 432 433 434 435 436 437 438 439 450
##  [75] 517 530 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 651
## [112] 712 732 750 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2008
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 365 366 367 368 369 370 371 372 373 402 403 404 411 420 432 433 434 435 436 437 438 439 450 451
##  [75] 517 530 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 651
## [112] 712 732 750 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2009
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 366 367 368 369 370 371 372 373 402 403 404 411 420 432 433 434 435 436 437 438 439 450 451 451
##  [75] 530 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 651 660
## [112] 732 750 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
```

```
## [1] 2010
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 366 367 368 369 370 371 372 373 402 403 404 411 420 432 433 434 435 436 437 438 439 450 451 461
##  [75] 531 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 651 660 663
## [112] 750 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2011
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 368 369 370 371 372 373 402 403 404 411 420 432 433 434 435 436 437 438 439 450 451 461 471
##  [75] 540 541 551 553 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 651 660 663 666
## [112] 760 770 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2012
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 368 369 370 371 372 373 402 403 404 411 420 432 433 434 435 436 437 438 439 450 451 461 471
##  [75] 541 551 553 560 565 570 571 572 580 581 590 600 615 616 625 630 640 645 651 660 663 666 670 679
## [112] 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2013
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 368 369 370 371 372 373 402 403 404 411 420 432 433 434 435 436 437 438 439 450 451 461 471
##  [75] 541 551 553 560 565 570 571 572 580 581 590 600 615 616 625 630 640 645 651 660 663 666 670 679
## [112] 771 775 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2014
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 140 145 150 155 160 165 290
##  [38] 369 370 371 372 373 402 403 404 411 420 432 433 434 435 437 438 439 450 451 452 461 471 475 481
##  [75] 553 560 565 570 571 572 580 581 590 600 615 616 625 630 640 645 651 660 663 666 670 679 690 694
## [112] 780 781 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
## [1] 2015
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 110 115 130 140 145 150 155 160 165 290 310
##  [38] 371 372 373 402 403 404 411 420 432 433 434 435 437 438 439 450 451 452 461 471 475 481 482 483
##  [75] 565 570 571 572 580 581 590 600 615 616 625 630 640 645 651 660 663 666 670 679 690 694 698 700
## [112] 790 800 811 812 820 830 840 850 910 935 940 950
## [1] All of your Countries are Matched.
```

```r
# Estimate the SAR model (may take a couple of minutes)
m_ratio_sar <- lagsarlm(
  dflfp_mlfp ~ flfp_mlfplag + dflfp_mlfplag +
    log_wdi_overvaluedlag + dlog_wdi_overvalued +
    log_gdpcaplag + dlog_gdpcap +
    sqlog_gdpcaplag + dsqlog_gdpcap +
    log_oilgascaplag + dlog_oilgascap +
    as_factor_irr_rate_regimelag +
    wdi_fertilitylag + dwdi_fertility +
    vdem_elec_demlag + dvdem_elec_dem +
    as_factor(cowcode),
  listw = mat2listw(W[[2]], style = "W"),
  Durbin = FALSE,
  data = df2
)

# View the results
```

```
summary(m_ratio_sar)
```

```
##
## Call:lagsarlm(formula = dflfp_mlfp ~ flfp_mlfplag + dflfp_mlfplag +     log_wdi_overvaluedlag + dlog
##     dlog_gdpcap + sqlog_gdpcaplag + dsqlog_gdpcap + log_oilgascaplag +     dlog_oilgascap + as_facto
##     dwdi_fertility + vdem_elec_demlag + dvdem_elec_dem + as_factor(cowcode),     data = df2, listw =
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -14.520699  -0.308596   0.010038   0.323290   9.140430
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##                                 Estimate Std. Error z value  Pr(>|z|)
## (Intercept)                    9.3686512  2.5813183  3.6294 0.0002841
## flfp_mlfplag                  -0.0517519  0.0056518 -9.1567 < 2.2e-16
## dflfp_mlfplag                  0.0710380  0.0177609  3.9997 6.343e-05
## log_wdi_overvaluedlag         -0.1706398  0.0650351 -2.6238 0.0086952
## dlog_wdi_overvalued           -0.2453671  0.0992232 -2.4729 0.0134029
## log_gdpcaplag                 -1.1841938  0.5963612 -1.9857 0.0470668
## dlog_gdpcap                    2.4834682  1.6379432  1.5162 0.1294659
## sqlog_gdpcaplag                0.0878457  0.0375218  2.3412 0.0192224
## dsqlog_gdpcap                 -0.0145274  0.1101067 -0.1319 0.8950324
## log_oilgascaplag              -0.0144295  0.0350525 -0.4117 0.6805922
## dlog_oilgascap                 0.0434567  0.0574349  0.7566 0.4492744
## as_factor_irr_rate_regimelag2 -0.0185251  0.0794629 -0.2331 0.8156615
## as_factor_irr_rate_regimelag3  0.0840937  0.0913404  0.9207 0.3572266
## as_factor_irr_rate_regimelag4 -0.1215208  0.1910420 -0.6361 0.5247146
## as_factor_irr_rate_regimelag5  0.1480148  0.1008032  1.4684 0.1420081
## as_factor_irr_rate_regimelag6 -0.0459340  0.2144539 -0.2142 0.8303984
## wdi_fertilitylag              -0.1356041  0.0515726 -2.6294 0.0085541
## dwdi_fertility                -1.0297885  0.4271567 -2.4108 0.0159177
## vdem_elec_demlag              -0.5861037  0.2277080 -2.5739 0.0100552
## dvdem_elec_dem                 0.0567370  0.4024256  0.1410 0.8878797
## as_factor(cowcode)42          -1.3323341  0.3696269 -3.6045 0.0003127
## as_factor(cowcode)51          -1.1704012  0.3477300 -3.3658 0.0007631
## as_factor(cowcode)52          -1.8114920  0.4504529 -4.0215 5.783e-05
## as_factor(cowcode)53          -0.6184865  0.4010174 -1.5423 0.1230023
## as_factor(cowcode)70          -2.0393350  0.4568960 -4.4635 8.065e-06
## as_factor(cowcode)90          -2.1641325  0.3753742 -5.7653 8.153e-09
## as_factor(cowcode)91          -1.8019439  0.3566528 -5.0524 4.363e-07
## as_factor(cowcode)92          -1.7016700  0.3676925 -4.6280 3.693e-06
## as_factor(cowcode)93          -1.6432843  0.3581341 -4.5885 4.465e-06
## as_factor(cowcode)94          -1.4654139  0.4298297 -3.4093 0.0006513
## as_factor(cowcode)95          -1.6144218  0.4077413 -3.9594 7.513e-05
## as_factor(cowcode)100         -1.0969324  0.4121594 -2.6614 0.0077810
## as_factor(cowcode)101         -1.7176259  0.4636821 -3.7043 0.0002120
## as_factor(cowcode)110         -2.1656339  0.3838849 -5.6414 1.687e-08
## as_factor(cowcode)115         -1.5398585  0.4287763 -3.5913 0.0003290
## as_factor(cowcode)130         -1.5611887  0.4351722 -3.5875 0.0003338
## as_factor(cowcode)135         -0.2219834  0.3458138 -0.6419 0.5209278
## as_factor(cowcode)140         -1.0915186  0.4197883 -2.6002 0.0093179
## as_factor(cowcode)145         -0.8446335  0.3336582 -2.5314 0.0113597
## as_factor(cowcode)150         -1.3118308  0.3386023 -3.8743 0.0001070
```

```
## as_factor(cowcode)155        -1.6000185  0.4529182 -3.5327 0.0004114
## as_factor(cowcode)160        -1.4810891  0.4096356 -3.6156 0.0002996
## as_factor(cowcode)165        -1.0966818  0.4098970 -2.6755 0.0074617
## as_factor(cowcode)290        -1.4188241  0.3876775 -3.6598 0.0002524
## as_factor(cowcode)310        -1.5702392  0.4001996 -3.9236 8.722e-05
## as_factor(cowcode)316        -1.6287479  0.4384103 -3.7151 0.0002031
## as_factor(cowcode)317        -1.9980561  0.4508716 -4.4315 9.356e-06
## as_factor(cowcode)339        -1.4160918  0.3387309 -4.1806 2.908e-05
## as_factor(cowcode)341        -1.0908621  0.4389083 -2.4854 0.0129406
## as_factor(cowcode)343        -1.7980626  0.3699337 -4.8605 1.171e-06
## as_factor(cowcode)344        -1.1108067  0.4132184 -2.6882 0.0071842
## as_factor(cowcode)345        -1.1585744  0.4316045 -2.6843 0.0072673
## as_factor(cowcode)346        -2.2035534  0.3920547 -5.6205 1.904e-08
## as_factor(cowcode)349        -2.2027841  0.5525735 -3.9864 6.708e-05
## as_factor(cowcode)352        -1.2758546  0.5312806 -2.4015 0.0163293
## as_factor(cowcode)355        -1.1935942  0.3421843 -3.4882 0.0004864
## as_factor(cowcode)359        -0.5938918  0.3115437 -1.9063 0.0566129
## as_factor(cowcode)360        -1.6773570  0.3662837 -4.5794 4.663e-06
## as_factor(cowcode)365        -1.2017633  0.4082554 -2.9437 0.0032436
## as_factor(cowcode)366        -1.2290699  0.4317368 -2.8468 0.0044161
## as_factor(cowcode)367        -0.9095618  0.4076884 -2.2310 0.0256797
## as_factor(cowcode)368        -0.9755488  0.3852461 -2.5323 0.0113325
## as_factor(cowcode)369        -1.2285812  0.3437078 -3.5745 0.0003509
## as_factor(cowcode)370        -1.0276805  0.3513101 -2.9253 0.0034415
## as_factor(cowcode)371        -1.3591659  0.3280597 -4.1430 3.427e-05
## as_factor(cowcode)372        -1.3361707  0.3331263 -4.0110 6.046e-05
## as_factor(cowcode)373        -0.1825724  0.3687773 -0.4951 0.6205472
## as_factor(cowcode)402        -1.0809293  0.3652020 -2.9598 0.0030783
## as_factor(cowcode)403        -0.3006232  0.5762216 -0.5217 0.6018691
## as_factor(cowcode)404        -0.1398119  0.3054795 -0.4577 0.6471822
## as_factor(cowcode)411        -0.3687177  0.4248195 -0.8679 0.3854273
## as_factor(cowcode)420        -0.4442126  0.3028930 -1.4666 0.1424941
## as_factor(cowcode)432        -0.5134717  0.3496525 -1.4685 0.1419630
## as_factor(cowcode)433        -0.9373249  0.3544478 -2.6445 0.0081820
## as_factor(cowcode)434         0.9984985  0.3198163  3.1221 0.0017957
## as_factor(cowcode)435        -1.7432937  0.3931998 -4.4336 9.267e-06
## as_factor(cowcode)436        -0.2669877  0.3555256 -0.7510 0.4526730
## as_factor(cowcode)437        -0.6460258  0.3460512 -1.8669 0.0619225
## as_factor(cowcode)438         0.4514212  0.3110692  1.4512 0.1467262
## as_factor(cowcode)439        -0.6057043  0.3210492 -1.8866 0.0592087
## as_factor(cowcode)450        -0.3422006  0.4361758 -0.7845 0.4327190
## as_factor(cowcode)451         0.5067282  0.3213302  1.5770 0.1148024
## as_factor(cowcode)452         0.3838154  0.3182788  1.2059 0.2278524
## as_factor(cowcode)461         0.5094560  0.3169190  1.6075 0.1079386
## as_factor(cowcode)471         0.1018684  0.3500004  0.2911 0.7710114
## as_factor(cowcode)475        -0.0830851  0.3740258 -0.2221 0.8242071
## as_factor(cowcode)481        -1.2936683  0.4450239 -2.9070 0.0036496
## as_factor(cowcode)482        -0.2911907  0.3066405 -0.9496 0.3423074
## as_factor(cowcode)483        -0.0764717  0.3488780 -0.2192 0.8264996
## as_factor(cowcode)484         0.3376846  0.3969636  0.8507 0.3949533
## as_factor(cowcode)490         0.5468329  0.3450436  1.5848 0.1130067
## as_factor(cowcode)500         0.1009507  0.3143695  0.3211 0.7481185
## as_factor(cowcode)501         0.0593727  0.2982144  0.1991 0.8421892
## as_factor(cowcode)510         0.2792699  0.3153140  0.8857 0.3757856
```

```
## as_factor(cowcode)516          0.8592214  0.3597544  2.3884 0.0169240
## as_factor(cowcode)517          0.4722736  0.3131863  1.5080 0.1315638
## as_factor(cowcode)530         -0.1039859  0.3292304 -0.3158 0.7521199
## as_factor(cowcode)531         -0.1624258  0.3453134 -0.4704 0.6380891
## as_factor(cowcode)540          0.3105315  0.4178940  0.7431 0.4574292
## as_factor(cowcode)541          1.1431820  0.3483309  3.2819 0.0010312
## as_factor(cowcode)551         -0.0092098  0.3214889 -0.0286 0.9771458
## as_factor(cowcode)553         -0.0525146  0.3119643 -0.1683 0.8663195
## as_factor(cowcode)560         -0.6952762  0.3843785 -1.8088 0.0704771
## as_factor(cowcode)565         -0.0997830  0.3468218 -0.2877 0.7735711
## as_factor(cowcode)570         -0.4863756  0.2960653 -1.6428 0.1004246
## as_factor(cowcode)571         -0.3485972  0.3517370 -0.9911 0.3216497
## as_factor(cowcode)572         -1.6891904  0.3758804 -4.4940 6.991e-06
## as_factor(cowcode)580          0.2056748  0.3068356  0.6703 0.5026604
## as_factor(cowcode)581         -0.6300731  0.3148616 -2.0011 0.0453804
## as_factor(cowcode)590         -1.7987073  0.4190665 -4.2922 1.769e-05
## as_factor(cowcode)600         -3.2537016  0.4345784 -7.4870 7.039e-14
## as_factor(cowcode)615         -3.8535530  0.5459003 -7.0591 1.676e-12
## as_factor(cowcode)616         -3.1854930  0.4678771 -6.8084 9.869e-12
## as_factor(cowcode)620         -3.4784179  0.5788459 -6.0092 1.864e-09
## as_factor(cowcode)625         -3.1643331  0.4390954 -7.2065 5.742e-13
## as_factor(cowcode)630         -3.9762868  0.5732198 -6.9368 4.012e-12
## as_factor(cowcode)640         -3.2737891  0.4930873 -6.6394 3.150e-11
## as_factor(cowcode)645         -3.2058368  0.5783329 -5.5432 2.969e-08
## as_factor(cowcode)651         -3.4513678  0.4735745 -7.2879 3.149e-13
## as_factor(cowcode)660         -3.5527422  0.5098043 -6.9688 3.196e-12
## as_factor(cowcode)663         -3.7717037  0.4934773 -7.6431 2.109e-14
## as_factor(cowcode)666         -0.8691275  0.4754560 -1.8280 0.0675515
## as_factor(cowcode)670         -4.2471722  0.6409800 -6.6261 3.448e-11
## as_factor(cowcode)679         -4.0086314  0.4861336 -8.2459 2.220e-16
## as_factor(cowcode)690         -2.9887980  0.6151304 -4.8588 1.181e-06
## as_factor(cowcode)694         -2.9607522  0.7257462 -4.0796 4.511e-05
## as_factor(cowcode)698         -3.7205010  0.6116742 -6.0825 1.183e-09
## as_factor(cowcode)700         -3.4412857  0.4887941 -7.0404 1.917e-12
## as_factor(cowcode)701         -2.8416382  0.7280588 -3.9030 9.499e-05
## as_factor(cowcode)702         -1.5719803  0.3130827 -5.0210 5.141e-07
## as_factor(cowcode)703         -1.5812625  0.3169101 -4.9896 6.050e-07
## as_factor(cowcode)704         -1.3821940  0.5085383 -2.7180 0.0065683
## as_factor(cowcode)705         -0.8420718  0.3786366 -2.2240 0.0261513
## as_factor(cowcode)710         -1.1972459  0.3365375 -3.5575 0.0003743
## as_factor(cowcode)712         -0.7005137  0.3120087 -2.2452 0.0247570
## as_factor(cowcode)732         -1.8701272  0.4616467 -4.0510 5.100e-05
## as_factor(cowcode)750         -2.9195212  0.4022654 -7.2577 3.937e-13
## as_factor(cowcode)760         -0.6802798  0.3088970 -2.2023 0.0276450
## as_factor(cowcode)770         -3.1081371  0.4561659 -6.8136 9.518e-12
## as_factor(cowcode)771         -2.6823201  0.4326066 -6.2004 5.633e-10
## as_factor(cowcode)775         -1.5416273  0.4222827 -3.6507 0.0002615
## as_factor(cowcode)780         -2.7773873  0.3792819 -7.3228 2.429e-13
## as_factor(cowcode)781         -1.7273447  0.4117011 -4.1956 2.721e-05
## as_factor(cowcode)790          0.0011585  0.2928015  0.0040 0.9968430
## as_factor(cowcode)800         -1.1468109  0.3352462 -3.4208 0.0006244
## as_factor(cowcode)811         -0.1749525  0.2996196 -0.5839 0.5592772
## as_factor(cowcode)812         -0.0867168  0.3091405 -0.2805 0.7790868
## as_factor(cowcode)820         -2.0834570  0.4198161 -4.9628 6.949e-07
```

```
## as_factor(cowcode)830         -2.1408179  0.5541432 -3.8633 0.0001119
## as_factor(cowcode)840         -1.3706512  0.3264311 -4.1989 2.682e-05
## as_factor(cowcode)850         -1.8004225  0.3659752 -4.9195 8.676e-07
## as_factor(cowcode)910          0.4786887  0.3411452  1.4032 0.1605627
## as_factor(cowcode)935         -0.6195914  0.3317197 -1.8678 0.0617877
## as_factor(cowcode)940         -0.3649198  0.3047138 -1.1976 0.2310797
## as_factor(cowcode)950         -2.1098355  0.3797078 -5.5565 2.753e-08
##
## Rho: 0.080519, LR test value: 3.5131, p-value: 0.060887
## Asymptotic standard error: 0.038475
##     z-value: 2.0928, p-value: 0.036367
## Wald statistic: 4.3798, p-value: 0.036367
##
## Log likelihood: -3949.02 for lag model
## ML residual variance (sigma squared): 0.84052, (sigma: 0.9168)
## Number of observations: 2964
## Number of parameters estimated: 157
## AIC: 8212, (AIC for lm: 8213.6)
## LM test for residual autocorrelation
## test value: 0.037146, p-value: 0.84717
```

```
# Table 2: Ratio ~ REER + Country Effects   | REER @ 80% HDI  ----------

m_ratio_strict <- plm(
  dflfp_mlfp ~ flfp_mlfplag + dflfp_mlfplag +
    log_wdi_overvaluedlag + dlog_wdi_overvalued +
    log_gdpcaplag + dlog_gdpcap +
    sqlog_gdpcaplag + dsqlog_gdpcap +
    log_oilgascaplag + dlog_oilgascap,
  data = subset(
    df_tscs,
    oecd == 0 & country != "Zimbabwe" &
      log_wdi_overvalued >= -.48 &
      log_wdi_overvalued <= .64
  ),
  model = "within",
  effect = "individual",
  index = c("cowcode", "year")
)

# View the results
show_pcse(m_ratio_strict)
```

```
## Oneway (individual) effect Within Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovBK(x, type = "HC1", cluster = "time")
##
## Call:
## plm(formula = dflfp_mlfp ~ flfp_mlfplag + dflfp_mlfplag + log_wdi_overvaluedlag +
##     dlog_wdi_overvalued + log_gdpcaplag + dlog_gdpcap + sqlog_gdpcaplag +
##     dsqlog_gdpcap + log_oilgascaplag + dlog_oilgascap, data = subset(df_tscs,
##     oecd == 0 & country != "Zimbabwe" & log_wdi_overvalued >=
##         -0.48 & log_wdi_overvalued <= 0.64), effect = "individual",
##     model = "within", index = c("cowcode", "year"))
##
```

```
## Unbalanced Panel: n = 137, T = 1-24, N = 2644
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -6.343428 -0.278981  0.004145  0.298747  8.162185
##
## Coefficients:
##                          Estimate Std. Error t-value  Pr(>|t|)
## flfp_mlfplag           -0.046223   0.010317 -4.4801 7.795e-06 ***
## dflfp_mlfplag           0.126158   0.052693  2.3942   0.01673 *
## log_wdi_overvaluedlag -0.203527   0.090531 -2.2482   0.02465 *
## dlog_wdi_overvalued   -0.405929   0.137570 -2.9507   0.00320 **
## log_gdpcaplag         -0.845552   0.464837 -1.8190   0.06903 .
## dlog_gdpcap           -2.018411   1.607041 -1.2560   0.20924
## sqlog_gdpcaplag         0.066860   0.030356  2.2026   0.02772 *
## dsqlog_gdpcap           0.233515   0.108483  2.1525   0.03145 *
## log_oilgascaplag      -0.017672   0.027192 -0.6499   0.51582
## dlog_oilgascap          0.025127   0.051814  0.4850   0.62775
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    2089.3
## Residual Sum of Squares: 1973.3
## R-Squared:      0.055491
## Adj. R-Squared: 0.00026569
## F-statistic: 5.71883 on 10 and 23 DF, p-value: 0.00026999
```

```
# Table 2: Ratio ~ REER + Extended Controls | REER @ 80% HDI  ----------

# Create the usable dataset
df <- df_tscs %>%
  filter(
    oecd == 0 &
      country_space != "Zimbabwe (Rhodesia)" &
      country_space != "Tonga" &
      log_wdi_overvalued >= -.48 &
      log_wdi_overvalued <= .64
  )

# Estimate the preliminary model
pre <- lm(
  dflfp_mlfp ~ flfp_mlfplag + dflfp_mlfplag +
    log_wdi_overvaluedlag + dlog_wdi_overvalued +
    log_gdpcaplag + dlog_gdpcap +
    sqlog_gdpcaplag + dsqlog_gdpcap +
    log_oilgascaplag + dlog_oilgascap +
    as_factor(irr_rate_regimelag) +
    wdi_fertilitylag + dwdi_fertility +
    vdem_elec_demlag + dvdem_elec_dem +
    as_factor(cowcode) - cowcode - year - country_space,
  data = df
)

# Make the data frame suitable for the SLX model
```

```r
df2 <- pre$model %>% data.frame() %>% clean_names()

# Make the spatial weights where k = 10
W <- make_ntspmat(pre, ci = country_space, y = year, k = 10)
```

```
## [1] 1992
##  [1] 42  51  52  70  90  91  93  94  95  100 101 110 135 140 145 150 155 165 290 310 355 411 432 434
## [38] 517 540 551 553 560 565 570 571 572 580 590 600 615 616 640 660 663 670 710 712 750 760 770 771
## [1] 1993
##  [1] 42  51  52  90  91  93  94  95  100 101 110 135 140 145 150 155 165 290 310 339 355 360 411 432
## [38] 530 540 551 553 560 565 570 571 572 580 590 600 615 616 640 651 660 663 670 710 750 760 770 771
## [1] All of your Countries are Matched.
## [1] 1994
##  [1] 42  51  52  90  91  93  94  95  100 101 110 135 140 145 150 290 310 316 339 355 360 369 411 432
## [38] 490 500 501 510 516 530 551 553 560 570 571 572 580 581 590 600 615 616 640 651 660 663 670 710
## [1] All of your Countries are Matched.
## [1] 1995
##  [1] 42  51  52  70  90  91  93  94  95  100 101 110 115 135 140 145 150 290 310 316 339 343 355 360
## [38] 471 475 481 483 484 490 500 501 510 516 530 540 551 553 560 565 570 571 572 580 581 590 600 615
## [75] 770 771 780 790 800 811 812 820 840 850 910
## [1] All of your Countries are Matched.
## [1] 1996
##  [1] 42  51  52  70  90  91  93  94  95  100 101 110 115 135 145 150 290 310 316 339 343 344 355 359
## [38] 461 471 481 483 484 490 500 501 510 516 517 530 540 551 553 560 565 570 571 572 580 581 590 600
## [75] 705 710 712 750 760 770 771 780 781 790 800 811 812 820 840 850 910 950
## [1] All of your Countries are Matched.
## [1] 1997
##  [1] 41  42  51  52  70  90  91  93  94  95  100 101 110 115 135 145 150 290 310 316 339 343 344 355
## [38] 437 438 439 451 452 461 471 481 482 483 484 490 500 501 510 516 530 540 551 553 560 565 570 571
## [75] 670 690 702 703 704 705 710 750 760 770 771 780 781 790 800 811 812 820 840 850 910
## [1] All of your Countries are Matched.
## [1] 1998
##  [1] 41  42  52  70  90  91  93  94  95  100 101 110 115 135 145 150 290 310 316 339 343 346 355 359
## [38] 438 439 451 452 461 471 481 483 484 490 500 501 510 516 530 531 540 551 553 560 565 570 571 572
## [75] 702 703 704 705 710 732 750 760 770 771 780 781 790 800 811 812 820 840 910 950
## [1] All of your Countries are Matched.
## [1] 1999
##  [1] 41  42  51  52  70  90  91  93  94  95  100 101 110 115 135 140 145 150 155 290 310 316 339 343
## [38] 404 432 433 434 436 437 438 439 451 452 461 471 481 482 483 484 500 501 510 516 517 530 551 553
## [75] 640 651 660 663 670 690 702 703 704 710 732 750 760 771 780 781 790 800 811 820 830 840 910 950
## [1] All of your Countries are Matched.
## [1] 2000
##  [1] 41  42  51  52  70  90  91  93  94  95  100 101 110 115 135 140 145 150 155 290 310 316 339 343
## [38] 433 434 436 437 438 439 451 461 471 481 482 483 484 500 501 510 516 517 530 540 551 553 560 565
## [75] 651 660 663 670 690 703 704 710 732 750 760 770 771 780 781 790 800 811 820 830 840 910 950
## [1] All of your Countries are Matched.
## [1] 2001
##  [1] 41  42  51  52  70  90  91  92  93  94  95  100 101 110 135 140 145 150 155 160 165 290 310 316
## [38] 420 432 433 434 436 437 438 439 451 461 471 481 482 483 484 500 501 510 516 517 530 540 551 553
## [75] 630 640 651 660 663 670 690 694 703 710 732 750 760 771 781 790 800 811 820 830 840 910 950
## [1] All of your Countries are Matched.
## [1] 2002
##  [1] 41  42  51  52  70  90  91  92  93  94  95  100 101 110 115 135 140 145 150 155 160 165 290 310
## [38] 402 404 420 432 433 434 436 437 438 439 451 461 471 481 482 483 484 490 500 501 510 516 517 530
```

```
## [75] 616 625 640 660 663 666 670 690 694 710 732 750 760 771 781 790 800 811 820 830 840 910 950
## [1] All of your Countries are Matched.
## [1] 2003
##   [1] 41  42  51  52  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165 290
##  [38] 372 402 404 420 432 433 434 436 437 438 439 451 461 471 481 482 483 484 490 500 501 510 516 517
##  [75] 600 616 625 640 660 663 666 670 690 694 700 703 710 732 750 760 771 781 790 800 811 820 830 840
## [1] All of your Countries are Matched.
## [1] 2004
##   [1] 41  42  51  52  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165 290
##  [38] 372 402 404 420 432 433 434 436 437 438 439 451 461 471 481 482 483 484 490 500 501 510 516 517
##  [75] 590 600 615 616 620 625 640 660 663 666 670 690 694 698 700 702 703 710 732 750 760 771 781 790
## [1] All of your Countries are Matched.
## [1] 2005
##   [1] 41  42  51  52  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165 290
##  [38] 371 372 402 404 420 432 433 434 436 437 438 439 451 461 471 481 482 483 484 490 500 501 510 516
##  [75] 590 600 615 616 620 625 640 660 663 666 670 690 694 698 700 702 703 710 732 750 760 771 781 790
## [1] All of your Countries are Matched.
## [1] 2006
##   [1] 41  42  51  52  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165 290
##  [38] 371 372 402 404 420 432 433 434 436 437 438 439 451 452 461 475 481 482 483 484 490 500 501 510
##  [75] 581 590 600 615 616 620 625 640 660 663 666 670 690 694 698 700 702 703 705 710 732 750 760 771
## [1] All of your Countries are Matched.
## [1] 2007
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 369 370 371 372 373 402 404 411 420 432 433 434 436 437 438 439 451 452 461 481 482 483 484 490
##  [75] 571 572 580 581 590 600 615 616 620 625 640 660 663 666 670 690 694 698 700 702 703 705 710 731
## [1] All of your Countries are Matched.
## [1] 2008
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 369 370 371 372 373 402 404 411 420 432 433 434 436 437 438 439 451 452 461 471 475 481 482 483
##  [75] 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 660 663 670 679 690 694 698 700
## [112] 820 830 840 850 910 940 950
## [1] All of your Countries are Matched.
## [1] 2009
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 369 370 371 372 373 402 404 411 420 432 433 434 436 437 438 439 451 452 461 471 481 482 483
##  [75] 560 565 570 571 572 580 581 590 600 616 620 625 630 640 660 663 666 670 690 694 698 700 702 703
## [112] 940 950
## [1] All of your Countries are Matched.
## [1] 2010
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 369 370 371 372 373 402 404 411 420 432 433 434 436 437 438 439 451 461 471 475 481 482 483
##  [75] 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 660 663 666 670 690 694 698 700
## [112] 811 820 830 840 850 910 940 950
## [1] All of your Countries are Matched.
## [1] 2011
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 369 370 371 372 373 402 404 411 420 432 433 434 436 437 438 439 451 461 471 475 481 482 483
##  [75] 560 565 570 571 572 580 581 590 600 615 616 620 625 630 640 645 660 663 666 670 679 690 694 698
## [112] 820 830 840 850 910 940 950
## [1] All of your Countries are Matched.
## [1] 2012
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 165
##  [38] 367 369 370 371 372 373 402 404 411 420 432 433 434 436 437 438 439 451 461 471 475 481 482 483
```

```
##  [75] 565 570 571 572 580 581 590 600 615 616 625 630 640 645 660 663 666 670 679 690 694 698 700 70
## [112] 820 830 840 850 910 940 950
## [1] All of your Countries are Matched.
## [1] 2013
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 135 140 145 150 155 160 16
##  [38] 367 369 370 371 372 373 402 404 411 420 432 433 434 436 437 438 439 451 461 471 475 481 482 48
##  [75] 565 570 571 572 580 581 590 600 615 616 625 630 640 645 660 663 666 670 679 690 694 698 700 70
## [112] 830 840 850 910 940 950
## [1] All of your Countries are Matched.
## [1] 2014
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 101 110 115 130 140 145 150 155 160 165 29
##  [38] 370 371 372 373 402 404 411 432 433 434 437 438 439 451 452 461 471 475 481 482 483 484 490 50
##  [75] 572 580 581 590 600 615 616 625 640 645 660 663 666 670 679 690 694 698 700 702 703 705 710 71
## [112] 940 950
## [1] All of your Countries are Matched.
## [1] 2015
##   [1] 41  42  51  52  53  70  90  91  92  93  94  95  100 110 115 130 140 145 150 155 160 165 290 31
##  [38] 372 402 404 411 420 432 433 434 437 438 439 451 452 461 471 475 481 482 483 490 500 501 510 51
##  [75] 590 600 616 625 640 645 651 660 663 666 670 679 690 694 698 700 702 703 705 710 712 732 760 77
## [1] All of your Countries are Matched.
```

```r
# Estimate the SAR model (may take a couple of minutes)
m_ratio_strict_sar <- lagsarlm(
  dflfp_mlfp ~ 0 + flfp_mlfplag + dflfp_mlfplag +
    log_wdi_overvaluedlag + dlog_wdi_overvalued +
    log_gdpcaplag + dlog_gdpcap +
    sqlog_gdpcaplag + dsqlog_gdpcap +
    log_oilgascaplag + dlog_oilgascap +
    as_factor_irr_rate_regimelag +
    wdi_fertilitylag + dwdi_fertility +
    vdem_elec_demlag + dvdem_elec_dem +
    as_factor(cowcode),
  listw = mat2listw(W[[2]], style = "W"),
  Durbin = FALSE,
  data = df2
)

# View the results
summary(m_ratio_strict_sar)
```

```
##
## Call:lagsarlm(formula = dflfp_mlfp ~ 0 + flfp_mlfplag + dflfp_mlfplag +     log_wdi_overvaluedlag + d
##     dlog_gdpcap + sqlog_gdpcaplag + dsqlog_gdpcap + log_oilgascaplag +     dlog_oilgascap + as_factor
##     dwdi_fertility + vdem_elec_demlag + dvdem_elec_dem + as_factor(cowcode),     data = df2, listw =
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -6.350790 -0.303700  0.015341  0.316495  8.233569
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##                                Estimate Std. Error z value  Pr(>|z|)
## flfp_mlfplag                 -0.0539580  0.0060941 -8.8541 < 2.2e-16
## dflfp_mlfplag                 0.0998069  0.0188789  5.2867 1.246e-07
## log_wdi_overvaluedlag        -0.1105454  0.1145803 -0.9648 0.3346527
```

```
## dlog_wdi_overvalued           -0.4007881  0.1582171 -2.5332 0.0113042
## log_gdpcaplag                 -1.9027795  0.7380203 -2.5782 0.0099310
## dlog_gdpcap                   -3.6554279  2.2972762 -1.5912 0.1115644
## sqlog_gdpcaplag                0.1358406  0.0461723  2.9420 0.0032606
## dsqlog_gdpcap                  0.3547347  0.1464598  2.4221 0.0154327
## log_oilgascaplag              -0.0222290  0.0397815 -0.5588 0.5763137
## dlog_oilgascap                 0.0218278  0.0647969  0.3369 0.7362182
## as_factor_irr_rate_regimelag1 12.3664534  3.1878682  3.8792 0.0001048
## as_factor_irr_rate_regimelag2 12.3455308  3.1920542  3.8676 0.0001099
## as_factor_irr_rate_regimelag3 12.5129525  3.1857953  3.9277 8.575e-05
## as_factor_irr_rate_regimelag4 12.3911067  3.1903836  3.8839 0.0001028
## as_factor_irr_rate_regimelag5 12.5985706  3.1932469  3.9454 7.967e-05
## as_factor_irr_rate_regimelag6 12.6073524  3.2094728  3.9282 8.559e-05
## wdi_fertilitylag              -0.2132820  0.0610683 -3.4925 0.0004785
## dwdi_fertility                -1.0441057  0.4634748 -2.2528 0.0242732
## vdem_elec_demlag              -0.4130923  0.2443445 -1.6906 0.0909106
## dvdem_elec_dem                 0.1998917  0.4222594  0.4734 0.6359378
## as_factor(cowcode)42          -1.4804126  0.3760576 -3.9367 8.262e-05
## as_factor(cowcode)51          -1.2485984  0.3562202 -3.5051 0.0004564
## as_factor(cowcode)52          -2.1590282  0.4681478 -4.6119 3.991e-06
## as_factor(cowcode)53          -0.9279296  0.4847259 -1.9143 0.0555769
## as_factor(cowcode)70          -2.3833073  0.4762072 -5.0048 5.593e-07
## as_factor(cowcode)90          -2.1155940  0.3785809 -5.5882 2.294e-08
## as_factor(cowcode)91          -1.8077291  0.3593961 -5.0299 4.907e-07
## as_factor(cowcode)92          -1.8171830  0.3962738 -4.5857 4.525e-06
## as_factor(cowcode)93          -1.7202095  0.3605056 -4.7717 1.827e-06
## as_factor(cowcode)94          -1.7885509  0.4440203 -4.0281 5.623e-05
## as_factor(cowcode)95          -1.8389498  0.4200339 -4.3781 1.197e-05
## as_factor(cowcode)100         -1.3594631  0.4220742 -3.2209 0.0012778
## as_factor(cowcode)101         -1.9627126  0.4792076 -4.0957 4.208e-05
## as_factor(cowcode)110         -2.2649524  0.3906285 -5.7982 6.702e-09
## as_factor(cowcode)115         -1.5528867  0.4428982 -3.5062 0.0004546
## as_factor(cowcode)130         -1.6337015  0.4410028 -3.7045 0.0002118
## as_factor(cowcode)135         -0.3584836  0.3488653 -1.0276 0.3041521
## as_factor(cowcode)140         -1.5228308  0.4408847 -3.4540 0.0005523
## as_factor(cowcode)145         -0.7710487  0.3355492 -2.2979 0.0215692
## as_factor(cowcode)150         -1.3697784  0.3415558 -4.0104 6.061e-05
## as_factor(cowcode)155         -1.8371097  0.4786781 -3.8379 0.0001241
## as_factor(cowcode)160         -1.9411575  0.4429441 -4.3824 1.174e-05
## as_factor(cowcode)165         -1.5849341  0.4404231 -3.5987 0.0003199
## as_factor(cowcode)290         -1.8100843  0.4023356 -4.4989 6.829e-06
## as_factor(cowcode)310         -1.9411215  0.4158702 -4.6676 3.047e-06
## as_factor(cowcode)316         -2.1177360  0.4605459 -4.5983 4.259e-06
## as_factor(cowcode)339         -1.4782777  0.3402407 -4.3448 1.394e-05
## as_factor(cowcode)341         -1.2837252  0.4416142 -2.9069 0.0036504
## as_factor(cowcode)343         -1.9798297  0.3764028 -5.2599 1.442e-07
## as_factor(cowcode)344         -1.4436157  0.4326819 -3.3364 0.0008486
## as_factor(cowcode)345         -1.3536252  0.4309252 -3.1412 0.0016825
## as_factor(cowcode)346         -2.3085395  0.4011411 -5.7549 8.668e-09
## as_factor(cowcode)355         -1.3879132  0.3459560 -4.0118 6.025e-05
## as_factor(cowcode)359         -0.6126758  0.3058268 -2.0033 0.0451405
## as_factor(cowcode)360         -1.8706116  0.3783552 -4.9441 7.651e-07
## as_factor(cowcode)365         -1.4094883  0.4239234 -3.3249 0.0008846
## as_factor(cowcode)367         -1.4097569  0.4485886 -3.1426 0.0016743
```

```
## as_factor(cowcode)369        -1.4544611  0.3654311 -3.9801 6.888e-05
## as_factor(cowcode)370        -1.1360590  0.3594703 -3.1604 0.0015757
## as_factor(cowcode)371        -1.4172232  0.3324988 -4.2623 2.023e-05
## as_factor(cowcode)372        -1.4097927  0.3327968 -4.2362 2.273e-05
## as_factor(cowcode)373        -0.7626556  0.4219778 -1.8073 0.0707099
## as_factor(cowcode)402        -1.0090334  0.3906916 -2.5827 0.0098035
## as_factor(cowcode)404         0.0440464  0.3145319  0.1400 0.8886299
## as_factor(cowcode)411        -0.3070530  0.4497690 -0.6827 0.4948024
## as_factor(cowcode)420        -0.2627868  0.3357796 -0.7826 0.4338521
## as_factor(cowcode)432        -0.3677034  0.3551252 -1.0354 0.3004732
## as_factor(cowcode)433        -0.8349250  0.3585186 -2.3288 0.0198686
## as_factor(cowcode)434         1.0996447  0.3217882  3.4173 0.0006325
## as_factor(cowcode)436        -0.0582720  0.3644275 -0.1599 0.8729598
## as_factor(cowcode)437        -0.4829012  0.3493355 -1.3823 0.1668666
## as_factor(cowcode)438         0.6962501  0.3221441  2.1613 0.0306722
## as_factor(cowcode)439        -0.3860280  0.3255882 -1.1856 0.2357675
## as_factor(cowcode)451         0.6282772  0.3222901  1.9494 0.0512459
## as_factor(cowcode)452         0.5590324  0.3428847  1.6304 0.1030212
## as_factor(cowcode)461         0.6704807  0.3160324  2.1216 0.0338750
## as_factor(cowcode)471         0.3647942  0.3609391  1.0107 0.3121693
## as_factor(cowcode)475         0.5429292  0.4486886  1.2100 0.2262653
## as_factor(cowcode)481        -1.2884483  0.4585248 -2.8100 0.0049544
## as_factor(cowcode)482        -0.2918913  0.3209914 -0.9093 0.3631689
## as_factor(cowcode)483         0.2844494  0.3590661  0.7922 0.4282486
## as_factor(cowcode)484         0.6393931  0.4216662  1.5163 0.1294312
## as_factor(cowcode)490         0.7204881  0.3577262  2.0141 0.0440014
## as_factor(cowcode)500         0.3114347  0.3153437  0.9876 0.3233466
## as_factor(cowcode)501         0.2556960  0.2990263  0.8551 0.3924984
## as_factor(cowcode)510         0.4977182  0.3173191  1.5685 0.1167622
## as_factor(cowcode)516         1.0289938  0.3719424  2.7665 0.0056654
## as_factor(cowcode)517         0.6155857  0.3231811  1.9048 0.0568100
## as_factor(cowcode)530         0.0506995  0.3402031  0.1490 0.8815321
## as_factor(cowcode)531        -0.1012292  0.4419734 -0.2290 0.8188386
## as_factor(cowcode)540         0.6643178  0.4509371  1.4732 0.1406987
## as_factor(cowcode)541         1.5111902  0.3795901  3.9811 6.859e-05
## as_factor(cowcode)551         0.1583764  0.3281561  0.4826 0.6293618
## as_factor(cowcode)553         0.1204926  0.3105866  0.3880 0.6980517
## as_factor(cowcode)560        -1.0135296  0.3993161 -2.5382 0.0111436
## as_factor(cowcode)565        -0.1270699  0.3581008 -0.3548 0.7227065
## as_factor(cowcode)570        -0.4095523  0.2911009 -1.4069 0.1594545
## as_factor(cowcode)571        -0.4693552  0.3575640 -1.3126 0.1893021
## as_factor(cowcode)572        -1.6176768  0.3835020 -4.2182 2.463e-05
## as_factor(cowcode)580         0.2663578  0.3045727  0.8745 0.3818300
## as_factor(cowcode)581        -0.5074831  0.3140840 -1.6158 0.1061471
## as_factor(cowcode)590        -2.1099337  0.4334417 -4.8679 1.128e-06
## as_factor(cowcode)600        -3.3722535  0.4479056 -7.5289 5.107e-14
## as_factor(cowcode)615        -3.9361174  0.5717771 -6.8840 5.819e-12
## as_factor(cowcode)616        -3.2876665  0.4831546 -6.8046 1.013e-11
## as_factor(cowcode)620        -3.6488654  0.6120456 -5.9618 2.495e-09
## as_factor(cowcode)625        -2.9794586  0.4485660 -6.6422 3.091e-11
## as_factor(cowcode)630        -4.2944293  0.6233404 -6.8894 5.604e-12
## as_factor(cowcode)640        -3.6313634  0.5157077 -7.0415 1.902e-12
## as_factor(cowcode)645        -3.2237242  0.6256515 -5.1526 2.569e-07
## as_factor(cowcode)651        -3.5276312  0.5287316 -6.6719 2.526e-11
```

```
## as_factor(cowcode)660         -3.8315472  0.5354126 -7.1563 8.291e-13
## as_factor(cowcode)663         -3.7728104  0.5109546 -7.3838 1.539e-13
## as_factor(cowcode)666         -1.4643836  0.5295893 -2.7651 0.0056900
## as_factor(cowcode)670         -4.4331052  0.6692800 -6.6237 3.503e-11
## as_factor(cowcode)679         -4.2001487  0.5841363 -7.1904 6.461e-13
## as_factor(cowcode)690         -3.4348283  0.6574303 -5.2246 1.745e-07
## as_factor(cowcode)694         -3.5401791  0.7883501 -4.4906 7.102e-06
## as_factor(cowcode)698         -3.8938692  0.6439504 -6.0468 1.477e-09
## as_factor(cowcode)700         -3.3009611  0.5039196 -6.5506 5.732e-11
## as_factor(cowcode)702         -1.6382649  0.3333209 -4.9150 8.879e-07
## as_factor(cowcode)703         -1.5910268  0.3186820 -4.9925 5.960e-07
## as_factor(cowcode)704         -1.3905281  0.5067090 -2.7442 0.0060652
## as_factor(cowcode)705         -0.8918326  0.4207472 -2.1196 0.0340364
## as_factor(cowcode)710         -1.1602771  0.3405716 -3.4069 0.0006572
## as_factor(cowcode)712         -0.8305413  0.3798561 -2.1865 0.0287818
## as_factor(cowcode)732         -2.5710173  0.5170246 -4.9727 6.602e-07
## as_factor(cowcode)750         -2.7793916  0.4149314 -6.6984 2.107e-11
## as_factor(cowcode)760          0.0750973  0.3091703  0.2429 0.8080833
## as_factor(cowcode)770         -3.3326403  0.5300214 -6.2877 3.221e-10
## as_factor(cowcode)771         -2.8036731  0.4493942 -6.2388 4.410e-10
## as_factor(cowcode)775         -3.2273877  1.1536459 -2.7976 0.0051491
## as_factor(cowcode)780         -2.8650871  0.4162834 -6.8825 5.880e-12
## as_factor(cowcode)781         -1.9096408  0.4217710 -4.5277 5.964e-06
## as_factor(cowcode)790          0.0991145  0.2910165  0.3406 0.7334196
## as_factor(cowcode)800         -1.2605674  0.3368352 -3.7424 0.0001823
## as_factor(cowcode)811         -0.0890504  0.2931832 -0.3037 0.7613288
## as_factor(cowcode)812          0.2973293  0.3668037  0.8106 0.4175982
## as_factor(cowcode)820         -2.2213868  0.4290626 -5.1773 2.251e-07
## as_factor(cowcode)830         -2.8191796  0.6171600 -4.5680 4.924e-06
## as_factor(cowcode)840         -1.3829162  0.3254169 -4.2497 2.141e-05
## as_factor(cowcode)850         -1.7883533  0.4101079 -4.3607 1.297e-05
## as_factor(cowcode)910          0.6583437  0.3537994  1.8608 0.0627749
## as_factor(cowcode)935         -0.9634633  0.9321877 -1.0336 0.3013462
## as_factor(cowcode)940         -0.4064394  0.3478829 -1.1683 0.2426768
## as_factor(cowcode)950         -2.2070567  0.3940318 -5.6012 2.129e-08
##
## Rho: 0.10659, LR test value: 5.5443, p-value: 0.018541
## Asymptotic standard error: 0.041781
##     z-value: 2.5513, p-value: 0.010732
## Wald statistic: 6.5092, p-value: 0.010732
##
## Log likelihood: -3154.237 for lag model
## ML residual variance (sigma squared): 0.7845, (sigma: 0.88572)
## Number of observations: 2430
## Number of parameters estimated: 148
## AIC: 6604.5, (AIC for lm: 6608)
## LM test for residual autocorrelation
## test value: 7.8528, p-value: 0.0050743
```

```r
# Table 2: Make the table ----------------------------------------------------

# Sanity test. Do my custom extract functions extract the right standard errors
# and p-values
coeftest(m_flfp_base, vcov. = vcovBK(m_flfp_base, type = "HC1", cluster = "time"))
```

```
##
## t test of coefficients:
##
##                         Estimate Std. Error t value  Pr(>|t|)
## flflplag               -0.048426   0.011129 -4.3516 1.396e-05 ***
## dflflplag               0.151854   0.056170  2.7035 0.0068992 **
## log_wdi_overvaluedlag -0.118229   0.035552 -3.3255 0.0008929 ***
## dlog_wdi_overvalued   -0.149872   0.053392 -2.8070 0.0050314 **
## log_gdpcaplag         -1.433910   0.343414 -4.1755 3.056e-05 ***
## dlog_gdpcap           -2.229163   0.905063 -2.4630 0.0138325 *
## sqlog_gdpcaplag        0.101055   0.022521  4.4871 7.483e-06 ***
## dsqlog_gdpcap          0.184915   0.066242  2.7915 0.0052788 **
## log_oilgascaplag      -0.014387   0.017340 -0.8297 0.4067880
## dlog_oilgascap         0.046859   0.038711  1.2105 0.2261871
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
show_pcse(m_flfp_base)
```

```
## Oneway (individual) effect Within Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovBK(x, type = "HC1", cluster = "time")
##
## Call:
## plm(formula = dflfp ~ flflplag + dflflplag + log_wdi_overvaluedlag +
##     dlog_wdi_overvalued + log_gdpcaplag + dlog_gdpcap + sqlog_gdpcaplag +
##     dsqlog_gdpcap + log_oilgascaplag + dlog_oilgascap, data = subset(df_tscs,
##     oecd == 0 & country != "Zimbabwe"), effect = "individual",
##     model = "within", index = c("cowcode", "year"))
##
## Unbalanced Panel: n = 147, T = 5-24, N = 3245
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -6.5269712 -0.2033263  0.0048274  0.2116495  6.4088103
##
## Coefficients:
##                         Estimate Std. Error t-value  Pr(>|t|)
## flflplag               -0.048426   0.011129 -4.3516 1.396e-05 ***
## dflflplag               0.151854   0.056170  2.7035 0.0068992 **
## log_wdi_overvaluedlag -0.118229   0.035552 -3.3255 0.0008929 ***
## dlog_wdi_overvalued   -0.149872   0.053392 -2.8070 0.0050314 **
## log_gdpcaplag         -1.433910   0.343414 -4.1755 3.056e-05 ***
## dlog_gdpcap           -2.229163   0.905063 -2.4630 0.0138325 *
## sqlog_gdpcaplag        0.101055   0.022521  4.4871 7.483e-06 ***
## dsqlog_gdpcap          0.184915   0.066242  2.7915 0.0052788 **
## log_oilgascaplag      -0.014387   0.017340 -0.8297 0.4067880
## dlog_oilgascap         0.046859   0.038711  1.2105 0.2261871
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    1867
## Residual Sum of Squares: 1760.6
## R-Squared:      0.056984
## Adj. R-Squared: 0.0093446
```

```
## F-statistic: 6.99925 on 10 and 23 DF, p-value: 5.9489e-05
```

```
data.frame("pcse" = extract_pcse(m_flfp_base),
           "pcse_sig" = extract_pcse_sig(m_flfp_base))
```

```
##                              pcse     pcse_sig
## flfplag               0.01112852 1.395529e-05
## dflfplag              0.05616961 6.899150e-03
## log_wdi_overvaluedlag 0.03555198 8.928771e-04
## dlog_wdi_overvalued   0.05339171 5.031398e-03
## log_gdpcaplag         0.34341373 3.056079e-05
## dlog_gdpcap           0.90506313 1.383250e-02
## sqlog_gdpcaplag       0.02252120 7.483227e-06
## dsqlog_gdpcap         0.06624242 5.278811e-03
## log_oilgascaplag      0.01734008 4.067880e-01
## dlog_oilgascap        0.03871126 2.261871e-01
```

```
screenreg(
  list(
    m_flfp_base,
    m_flfp_sar,
    m_flfp_strict,
    m_flfp_strict_sar,
    m_ratio_base,
    m_ratio_sar,
    m_ratio_strict,
    m_ratio_strict_sar
  ),
  override.se = list(
    extract_pcse(m_flfp_base),
    m_flfp_sar,
    extract_pcse(m_flfp_strict),
    m_flfp_strict_sar,
    extract_pcse(m_ratio_base),
    m_ratio_sar,
    extract_pcse(m_ratio_strict),
    m_ratio_strict_sar
  ),
  override.pvalues = list(
    extract_pcse_sig(m_flfp_base),
    m_flfp_sar,
    extract_pcse_sig(m_flfp_strict),
    m_flfp_strict_sar,
    extract_pcse_sig(m_ratio_base),
    m_ratio_sar,
    extract_pcse_sig(m_ratio_strict),
    m_ratio_strict_sar
  ),
  # Add significance stars
  stars = c(0.01, 0.05, 0.1),
  # Add a header
  custom.header = list(
    "$\\Delta$FLFP" = 1:4,
    "$\\Delta$Labor Force Ratio" = 5:8
  ),
```

```r
  # Variable names
  custom.coef.map = list_var_names_texreg,
  # Goodness-of-fit statistics
  include.rsquared = FALSE,
  include.adj = TRUE,
  include.adjrs = FALSE,
  include.aic = FALSE,
  include.loglik = FALSE,
  include.lr = FALSE,
  # Custom notes
  custom.note = "%stars,
  All models include country-specific fixed effects",
)
```

```
## Warning in override(models = models, override.coef = override.coef, override.se = override.se, : Numl
## coefficients in model 2. Using default SEs.

## Warning in override(models = models, override.coef = override.coef, override.se = override.se, : Numl
## coefficients in model 2. Using default p-values.

## Warning in override(models = models, override.coef = override.coef, override.se = override.se, : Numl
## coefficients in model 4. Using default SEs.

## Warning in override(models = models, override.coef = override.coef, override.se = override.se, : Numl
## coefficients in model 4. Using default p-values.

## Warning in override(models = models, override.coef = override.coef, override.se = override.se, : Numl
## coefficients in model 6. Using default SEs.

## Warning in override(models = models, override.coef = override.coef, override.se = override.se, : Numl
## coefficients in model 6. Using default p-values.

## Warning in override(models = models, override.coef = override.coef, override.se = override.se, : Numl
## coefficients in model 8. Using default SEs.

## Warning in override(models = models, override.coef = override.coef, override.se = override.se, : Numl
## coefficients in model 8. Using default p-values.

##
## ==========================================================================================
##                                                                $\Delta$FLFP
##                                                  ------------------------------------------------  ----
##                                                  Model 1      Model 2      Model 3      Model 4      Mode
## ------------------------------------------------------------------------------------------------------
## LDV$_{(t-1)}$                                     -0.05 ***    -0.05 ***    -0.05 ***    -0.05 ***    -(
##                                                   (0.01)       (0.01)       (0.01)       (0.01)       ((
## $\Delta$LDV$_{(t-1)}$                              0.15 ***     0.12 ***     0.15 **      0.12 ***     (
##                                                   (0.06)       (0.02)       (0.06)       (0.02)       ((
## logOvervalued$_{(t-1)}$                           -0.12 ***    -0.12 **     -0.20 **     -0.18 *      -(
##                                                   (0.04)       (0.05)       (0.08)       (0.10)       ((
## $\Delta$logOvervalued$_t$                         -0.15 ***    -0.16 *      -0.34 ***    -0.36 ***    -(
##                                                   (0.05)       (0.08)       (0.12)       (0.14)       ((
## GDP$_{(t-1)}$                                     -1.43 ***    -1.50 ***    -1.35 ***    -1.85 ***    -(
##                                                   (0.34)       (0.48)       (0.40)       (0.63)       ((
## $\Delta$GDP$_t$                                   -2.23 **     -2.77 **     -4.93 ***    -6.25 ***     3
##                                                   (0.91)       (1.35)       (1.52)       (2.02)       (1
## GDP$^{2}_{(t-1)}$                                  0.10 ***     0.11 ***     0.10 ***     0.13 ***     (
##                                                   (0.02)       (0.03)       (0.03)       (0.04)       ((
```

```
## $\Delta$GDP$^{2}_t$                                0.18 ***     0.23 **      0.35 ***     0.45 ***    -0
##                                                   (0.07)       (0.09)       (0.10)       (0.13)       (0
## Resource Rents$_{(t-1)}$                          -0.01        -0.03        -0.02        -0.03         0
##                                                   (0.02)       (0.03)       (0.02)       (0.04)       (0
## $\Delta$Resource Rents$_t$                         0.05         0.03         0.04         0.02         0
##                                                   (0.04)       (0.05)       (0.05)       (0.06)       (0
## Fertility$_{(t-1)}$                                            -0.05                     -0.09 *
##                                                                (0.04)                    (0.05)
## $\Delta$Fertility$_t$                                          -0.02                     -0.12
##                                                                (0.35)                    (0.41)
## Regime Type$_{(t-1)}$                                          -0.51 ***                 -0.53 **
##                                                                (0.19)                    (0.21)
## $\Delta$Regime Type$_t$                                         0.06                      0.06
##                                                                (0.33)                    (0.37)
## Exchange Rate: Narrow Crawling$_t$                             -0.09                     -0.11
##                                                                (0.07)                    (0.08)
## Exchange Rate: Wide Crawling$_t$                               -0.05                     -0.02
##                                                                (0.08)                    (0.09)
## Exchange Rate: Freely Floating$_t$                             -0.00                      0.04
##                                                                (0.16)                    (0.19)
## Exchange Rate: Freely Falling$_t$                               0.03                      0.04
##                                                                (0.08)                    (0.10)
## Exchange Rate: Dual Market$_t$                                 -0.14                      0.00
##                                                                (0.18)                    (0.25)
## Spatial Lag: $\Delta$Dependent Variable$_t$                     0.11 ***                  0.06
##                                                                (0.04)                    (0.04)
## ------------------------------------------------------------------------------------------------------
## Num. obs.                                        3245         2964         2644         2430         324
## Parameters                                                     157                       148
## ======================================================================================================
## *** p < 0.01; ** p < 0.05; * p < 0.1,
##   All models include country-specific fixed effects
```

```
# LaTeX Code
# texreg(
#   list(
#     m_flfp_base,
#     m_flfp_sar,
#     m_flfp_strict,
#     m_flfp_strict_sar,
#     m_ratio_base,
#     m_ratio_sar,
#     m_ratio_strict,
#     m_ratio_strict_sar
#   ),
#   override.se = list(
#     extract_pcse(m_flfp_base),
#     m_flfp_sar,
#     extract_pcse(m_flfp_strict),
#     m_flfp_strict_sar,
#     extract_pcse(m_ratio_base),
#     m_ratio_sar,
#     extract_pcse(m_ratio_strict),
#     m_ratio_strict_sar
```

```
#   ),
#   override.pvalues = list(
#     extract_pcse_sig(m_flfp_base),
#     m_flfp_sar,
#     extract_pcse_sig(m_flfp_strict),
#     m_flfp_strict_sar,
#     extract_pcse_sig(m_ratio_base),
#     m_ratio_sar,
#     extract_pcse_sig(m_ratio_strict),
#     m_ratio_strict_sar
#   ),
#   # Add significance stars
#   stars = c(0.01, 0.05, 0.1),
#   # Add a header
#   custom.header = list(
#     "$\\Delta$FLFP" = 1:4,
#     "$\\Delta$Labor Force Ratio" = 5:8
#   ),
#   # Variable names
#   custom.coef.map = list_var_names_texreg,
#   # Goodness-of-fit statistics
#   include.rsquared = FALSE,
#   include.adj = TRUE,
#   include.adjrs = FALSE,
#   include.aic = FALSE,
#   include.loglik = FALSE,
#   include.lr = FALSE,
#   # Custom notes
#   custom.note = "%stars,
#   All models include country-specific fixed effects",
#   # LaTeX formatting options
#   booktabs = TRUE,
#   fontsize = "scriptsize",
#   siunitx = TRUE,
#   threeparttable = FALSE,
#   sideways = TRUE,
#   float.pos = "tp",
#   label = "tab:results"
# )

######################################################################
##  Table 3: Female Employment ~ Industry | country == Mexico  ##
######################################################################


# Table 3: Make data: ILO | Mexico ------------------------------

# First, make a dataset of ILO variables just for the case of Mexico.
# Also, create logged and lagged values
df_ilo_ind_mex <- df_ilo_ind %>%
  filter(country == "Mexico") %>%
  arrange(industry, year) %>%
  mutate(log_employment = log(employment)) %>%
```

```
  group_by(industry) %>%
  mutate(log_employmentlag = dplyr::lag(log_employment, 1)) %>%
  ungroup()
# Table 3: Female Employment ~ ILO Industry Fixed Effects | country == "Mexico" ---------

# Make a panel object for use with `feols`
pdat <- panel(df_ilo_ind_mex, ~ industry + year)

# Estimate the model. Note: Manufacturing is the excluded category
m_emp_ind_mex <- feols(log_employment ~ industry, data = pdat)

# Footnote 10 refers to various robustness checks. Estimate them here.
m_emp_ind_mex_footnote10_v1 <-
  feols(log_employment ~ industry + l(log_employment, 1:2), data = pdat)
```

```
## NOTE: 44 observations removed because of NA values (RHS: 44).
```

```
m_emp_ind_mex_footnote10_v2 <-
  feols(log_employment ~ industry + year + l(log_employment, 1:2), data = pdat)
```

```
## NOTE: 44 observations removed because of NA values (RHS: 44).
```

```
m_emp_ind_mex_footnote10_v3 <-
  feols(log_employment ~ industry + factor(year) + l(log_employment, 1:2),
        data = pdat)
```

```
## NOTE: 44 observations removed because of NA values (RHS: 44).
## The variable 'factor(year)2021' has been removed because of collinearity (see $collin.var).
```

```
# View the results with clustered standard errors
summary(m_emp_ind_mex, cluster = c("industry", "year"))
```

```
## Variance contained negative values in the diagonal and was 'fixed' (a la Cameron, Gelbach & Miller 2
```

```
## OLS estimation, Dep. Var.: log_employment
## Observations: 264
## Standard-errors: Clustered (industry & year)
##                                                Estimate Std. Error   t value   Pr(>|t|)
## (Intercept)                                    8.015407   0.025170  318.4486  < 2.2e-16 ***
## industryAgriculture; forestry and fishing     -1.408077   0.014933  -94.2910  < 2.2e-16 ***
## industryMining and quarrying                  -5.023125   0.045079 -111.4293  < 2.2e-16 ***
## industryConstruction                          -3.068306   0.018441 -166.3835  < 2.2e-16 ***
## industryWholesale and retail trade             0.510409   0.011025   46.2954 5.8457e-14 ***
## industryTransportation and storage            -2.867759   0.023969 -119.6426  < 2.2e-16 ***
## industryAccommodation and food service        -0.339608   0.011956  -28.4039 1.2079e-11 ***
## industryInformation and communication         -3.108741   0.027367 -113.5928  < 2.2e-16 ***
## industryReal estate activities                -3.644994   0.035030 -104.0536  < 2.2e-16 ***
## industryPublic administration                 -1.222560   0.021243  -57.5516 5.3840e-15 ***
## industryHealth and social work                -1.154788   0.011645  -99.1669  < 2.2e-16 ***
## industryArts, entertainment and recreation    -3.281452   0.039817  -82.4144  < 2.2e-16 ***
## industryActivities of household as employers  -0.387696   0.019990  -19.3943 7.4314e-10 ***
## industryExtraterritorial organizations        -7.791346   0.058629 -132.8913  < 2.2e-16 ***
## industryNot elsewhere classified              -2.777548   0.048429  -57.3530 5.5920e-15 ***
## industryEducation                             -0.640623   0.013881  -46.1496 6.0513e-14 ***
## industryOther services                        -1.442975   0.009895 -145.8326  < 2.2e-16 ***
## industryElectricity; gas, and steam           -4.932861   0.029936 -164.7808  < 2.2e-16 ***
## industryFinance and insurance                 -2.482025   0.011392 -217.8723  < 2.2e-16 ***
```

```
## industryScientific and technical activities    -1.764452   0.017584 -100.3416  < 2.2e-16 ***
## industryAdministrative and support services    -1.816268   0.006865 -264.5792  < 2.2e-16 ***
## industryWater supply; sewage, waste management -4.223428   0.019189 -220.0915  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 0.148473   Adj. R2: 0.993495
```

```
summary(m_emp_ind_mex_footnote10_v1, cluster = c("industry", "year"))
```

```
## OLS estimation, Dep. Var.: log_employment
## Observations: 220
## Standard-errors: Clustered (industry & year)
##                                                 Estimate Std. Error   t value  Pr(>|t|)
## (Intercept)                                     5.546901   1.883675  2.944722 0.0163607 *
## industryAgriculture; forestry and fishing      -0.982115   0.338133 -2.904520 0.0174664 *
## industryMining and quarrying                   -3.436878   1.189725 -2.888800 0.0179192 *
## industryConstruction                           -2.107493   0.726072 -2.902595 0.0175212 *
## industryWholesale and retail trade              0.334417   0.113924  2.935440 0.0166094 *
## industryTransportation and storage             -1.957269   0.670862 -2.917543 0.0171001 *
## industryAccommodation and food service         -0.231065   0.074799 -3.089148 0.0129473 *
## industryInformation and communication          -2.162874   0.736447 -2.936906 0.0165699 *
## industryReal estate activities                 -2.478579   0.858897 -2.885769 0.0180080 *
## industryPublic administration                  -0.856943   0.296310 -2.892047 0.0178247 *
## industryHealth and social work                 -0.791162   0.278298 -2.842863 0.0193127 *
## industryArts, entertainment and recreation     -2.304978   0.755899 -3.049321 0.0138081 *
## industryActivities of household as employers   -0.288233   0.085747 -3.361450 0.0083699 **
## industryExtraterritorial organizations         -5.434436   1.826087 -2.976001 0.0155502 *
## industryNot elsewhere classified               -1.963154   0.647187 -3.033365 0.0141693 *
## industryEducation                              -0.462694   0.151935 -3.045330 0.0138975 *
## industryOther services                         -0.987707   0.343087 -2.878880 0.0182112 *
## industryElectricity; gas, and steam            -3.409326   1.180872 -2.887125 0.0179682 *
## industryFinance and insurance                  -1.697376   0.592175 -2.866345 0.0185871 *
## industryScientific and technical activities    -1.205184   0.417058 -2.889728 0.0178922 *
## industryAdministrative and support services    -1.252474   0.427414 -2.930351 0.0167474 *
## industryWater supply; sewage, waste management -2.884327   0.995580 -2.897132 0.0176778 *
## l(log_employment, 1)                            0.142959   0.164318  0.870011 0.4068997
## l(log_employment, 2)                            0.168928   0.112313  1.504081 0.1668157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 0.133853   Adj. R2: 0.994593
```

```
summary(m_emp_ind_mex_footnote10_v1, cluster = c("industry", "year"))
```

```
## OLS estimation, Dep. Var.: log_employment
## Observations: 220
## Standard-errors: Clustered (industry & year)
##                                                 Estimate Std. Error   t value  Pr(>|t|)
## (Intercept)                                     5.546901   1.883675  2.944722 0.0163607 *
## industryAgriculture; forestry and fishing      -0.982115   0.338133 -2.904520 0.0174664 *
## industryMining and quarrying                   -3.436878   1.189725 -2.888800 0.0179192 *
## industryConstruction                           -2.107493   0.726072 -2.902595 0.0175212 *
## industryWholesale and retail trade              0.334417   0.113924  2.935440 0.0166094 *
## industryTransportation and storage             -1.957269   0.670862 -2.917543 0.0171001 *
## industryAccommodation and food service         -0.231065   0.074799 -3.089148 0.0129473 *
## industryInformation and communication          -2.162874   0.736447 -2.936906 0.0165699 *
```

```
## industryReal estate activities                 -2.478579   0.858897 -2.885769 0.0180080 *
## industryPublic administration                  -0.856943   0.296310 -2.892047 0.0178247 *
## industryHealth and social work                 -0.791162   0.278298 -2.842863 0.0193127 *
## industryArts, entertainment and recreation     -2.304978   0.755899 -3.049321 0.0138081 *
## industryActivities of household as employers   -0.288233   0.085747 -3.361450 0.0083699 **
## industryExtraterritorial organizations         -5.434436   1.826087 -2.976001 0.0155502 *
## industryNot elsewhere classified               -1.963154   0.647187 -3.033365 0.0141693 *
## industryEducation                              -0.462694   0.151935 -3.045330 0.0138975 *
## industryOther services                         -0.987707   0.343087 -2.878880 0.0182112 *
## industryElectricity; gas, and steam            -3.409326   1.180872 -2.887125 0.0179682 *
## industryFinance and insurance                  -1.697376   0.592175 -2.866345 0.0185871 *
## industryScientific and technical activities    -1.205184   0.417058 -2.889728 0.0178922 *
## industryAdministrative and support services    -1.252474   0.427414 -2.930351 0.0167474 *
## industryWater supply; sewage, waste management -2.884327   0.995580 -2.897132 0.0176778 *
## l(log_employment, 1)                            0.142959   0.164318  0.870011 0.4068997
## l(log_employment, 2)                            0.168928   0.112313  1.504081 0.1668157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 0.133853   Adj. R2: 0.994593
```

```
summary(m_emp_ind_mex_footnote10_v1, cluster = c("industry", "year"))
```

```
## OLS estimation, Dep. Var.: log_employment
## Observations: 220
## Standard-errors: Clustered (industry & year)
##                                                 Estimate Std. Error   t value  Pr(>|t|)
## (Intercept)                                     5.546901   1.883675  2.944722 0.0163607 *
## industryAgriculture; forestry and fishing      -0.982115   0.338133 -2.904520 0.0174664 *
## industryMining and quarrying                   -3.436878   1.189725 -2.888800 0.0179192 *
## industryConstruction                           -2.107493   0.726072 -2.902595 0.0175212 *
## industryWholesale and retail trade              0.334417   0.113924  2.935440 0.0166094 *
## industryTransportation and storage             -1.957269   0.670862 -2.917543 0.0171001 *
## industryAccommodation and food service         -0.231065   0.074799 -3.089148 0.0129473 *
## industryInformation and communication          -2.162874   0.736447 -2.936906 0.0165699 *
## industryReal estate activities                 -2.478579   0.858897 -2.885769 0.0180080 *
## industryPublic administration                  -0.856943   0.296310 -2.892047 0.0178247 *
## industryHealth and social work                 -0.791162   0.278298 -2.842863 0.0193127 *
## industryArts, entertainment and recreation     -2.304978   0.755899 -3.049321 0.0138081 *
## industryActivities of household as employers   -0.288233   0.085747 -3.361450 0.0083699 **
## industryExtraterritorial organizations         -5.434436   1.826087 -2.976001 0.0155502 *
## industryNot elsewhere classified               -1.963154   0.647187 -3.033365 0.0141693 *
## industryEducation                              -0.462694   0.151935 -3.045330 0.0138975 *
## industryOther services                         -0.987707   0.343087 -2.878880 0.0182112 *
## industryElectricity; gas, and steam            -3.409326   1.180872 -2.887125 0.0179682 *
## industryFinance and insurance                  -1.697376   0.592175 -2.866345 0.0185871 *
## industryScientific and technical activities    -1.205184   0.417058 -2.889728 0.0178922 *
## industryAdministrative and support services    -1.252474   0.427414 -2.930351 0.0167474 *
## industryWater supply; sewage, waste management -2.884327   0.995580 -2.897132 0.0176778 *
## l(log_employment, 1)                            0.142959   0.164318  0.870011 0.4068997
## l(log_employment, 2)                            0.168928   0.112313  1.504081 0.1668157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 0.133853   Adj. R2: 0.994593
```

```r
# Table 3: Make the table  ------------------------------------------------

# Create a list of variable names for the texreg table
list_ilo_var_names_texreg <- list(
  'industryAccommodation and food service'         = 'Accommodation and food service',
  'industryAdministrative and support services'    = 'Administrative and support services',
  'industryAgriculture; forestry and fishing'      = 'Agriculture; forestry and fishing',
  'industryArts, entertainment and recreation'     = 'Arts, entertainment and recreation',
  'industryConstruction'                           = 'Construction',
  'industryEducation'                              = 'Education',
  'industryElectricity; gas, and steam'            = 'Electricity; gas, and steam',
  'industryExtraterritorial organizations'         = 'Extraterritorial organizations',
  'industryFinance and insurance'                  = 'Finance and insurance',
  'industryHealth and social work'                 = 'Health and social work',
  'industryActivities of household as employers'   = 'Household as employers',
  'industryInformation and communication'          = 'Information and communication',
  'industryMining and quarrying'                   = 'Mining and quarrying',
  'industryNot elsewhere classified'               = 'Not elsewhere classified',
  'industryOther services'                         = 'Other services',
  'industryPublic administration'                  = 'Public administration',
  'industryReal estate activities'                 = 'Real estate activities',
  'industryScientific and technical activities'    = 'Scientific and technical activities',
  'industryTransportation and storage'             = 'Transportation and storage',
  'industryWater supply; sewage, waste management' = 'Water supply; sewage, waste management',
  "industryWholesale and retail trade"             = 'Wholesale and retail trade',
  'year'                                           = 'Time Trend',
  'log_employmentlag'                              = 'Female Employment (Logged)$_{(t-1)}$',
  '(Intercept)'                                    = 'Constant'
)

# make the table
screenreg(
  m_emp_ind_mex,
  # Get the standard errors and p-values right
  override.se = summary(m_emp_ind_mex, cluster = c("industry", "year"))$coeftable[, 2],
  override.pvalues = summary(m_emp_ind_mex, cluster = c("industry", "year"))$coeftable[, 4],
  single.row = TRUE,
  stars = c(0.01, 0.05, 0.1),
  caption = "",
  # Coefficient labels
  custom.coef.map = list_ilo_var_names_texreg,
  # Goodness-of-fit statistics
  include.rsquared = FALSE,
  include.adjrs = FALSE,
  include.loglik = FALSE,
  # Custom notes
  custom.note = "%stars,
  Standard errors clustered by industry and year in parentheses",
)
```

```
## Variance contained negative values in the diagonal and was 'fixed' (a la Cameron, Gelbach & Miller 20
## Variance contained negative values in the diagonal and was 'fixed' (a la Cameron, Gelbach & Miller 20
```

```
##
```

```
## ============================================================
##                                          Model 1
## ------------------------------------------------------------
## Accommodation and food service            -0.34 (0.01) ***
## Administrative and support services       -1.82 (0.01) ***
## Agriculture; forestry and fishing         -1.41 (0.01) ***
## Arts, entertainment and recreation        -3.28 (0.04) ***
## Construction                              -3.07 (0.02) ***
## Education                                 -0.64 (0.01) ***
## Electricity; gas, and steam               -4.93 (0.03) ***
## Extraterritorial organizations            -7.79 (0.06) ***
## Finance and insurance                     -2.48 (0.01) ***
## Health and social work                    -1.15 (0.01) ***
## Household as employers                    -0.39 (0.02) ***
## Information and communication             -3.11 (0.03) ***
## Mining and quarrying                      -5.02 (0.05) ***
## Not elsewhere classified                  -2.78 (0.05) ***
## Other services                            -1.44 (0.01) ***
## Public administration                     -1.22 (0.02) ***
## Real estate activities                    -3.64 (0.04) ***
## Scientific and technical activities       -1.76 (0.02) ***
## Transportation and storage                -2.87 (0.02) ***
## Water supply; sewage, waste management    -4.22 (0.02) ***
## Wholesale and retail trade                 0.51 (0.01) ***
## Constant                                   8.02 (0.03) ***
## ------------------------------------------------------------
## Num. obs.                                 264
## ============================================================
## *** p < 0.01; ** p < 0.05; * p < 0.1,
##   Standard errors clustered by industry and year in parentheses
```

```r
#######################################################################
##               Table 4: FLFP Models for Mexico                     ##
#######################################################################

# Table 4: Make data: FLFP ~ REER | Mexico ----------------------------------------

# Create a database of that just the case of Mexico since 1990.
df_mex_case <- df_tscs %>% filter(country == "Mexico" & year >= 1990)

# Table 4: FLFP ~ REER | Mexico ---------------------------------------------

# Estimate the ECM model via `dynardl`
m_mex_flfp <- dynardl(
  flfp ~ log_wdi_overvalued + log_gdpcap,
  data = df_mex_case,
  lags = list(
    "flfp" = 1,
    "log_wdi_overvalued" = 1,
    "log_gdpcap" = 1
  ),
  diffs = c("log_wdi_overvalued",
            "log_gdpcap"),
  lagdiffs = list("flfp" = 1:2),
```

```
  trend = FALSE,
  ec = TRUE,
  simulate = FALSE, # No need to simulate here
  fullsims = FALSE,
  shockvar = "log_wdi_overvalued",
  shockval = -(sd(df_mex_case$log_wdi_overvalued)),
  time = 2,
  range = 10
)
```

```
## [1] "Error correction (EC) specified; dependent variable to be run in differences."
```

```
# View the results
summary(m_mex_flfp)
```

```
##
## Call:
## lm(formula = as.formula(paste(paste(dvnamelist), "~", paste(colnames(IVs),
##     collapse = "+"), collapse = " ")))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08422 -0.45032  0.08105  0.30712  1.39302
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -108.3979    52.3036  -2.072  0.05587 .
## l.1.flfp                -0.8226     0.2441  -3.370  0.00421 **
## ld.1.flfp                0.2075     0.2252   0.921  0.37158
## ld.2.flfp                0.3544     0.2103   1.685  0.11261
## d.1.log_wdi_overvalued  -3.2538     2.6396  -1.233  0.23665
## l.1.log_wdi_overvalued  -5.6953     1.8037  -3.158  0.00650 **
## d.1.log_gdpcap           8.2351     9.8277   0.838  0.41521
## l.1.log_gdpcap          15.8282     6.7246   2.354  0.03264 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7065 on 15 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.5064, Adjusted R-squared:  0.2761
## F-statistic: 2.198 on 7 and 15 DF,  p-value: 0.0949
```

```
# Table 4: Labor Force Ratio ~ REER | Mexico ------------------------------------------------

# Estimate the ECM model via `dynardl`
m_mex_flfp_mlfp <- dynardl(
  flfp_mlfp ~ log_wdi_overvalued + log_gdpcap,
  data = df_mex_case,
  lags = list(
    "flfp_mlfp" = 1,
    "log_wdi_overvalued" = 1,
    "log_gdpcap" = 1
  ),
  diffs = c("log_wdi_overvalued",
            "log_gdpcap"),
```

```
  lagdiffs = list("flfp_mlfp" = 1:2),
  trend = FALSE,
  ec = TRUE,
  simulate = FALSE,
  fullsims = FALSE,
  shockvar = "log_wdi_overvalued",
  shockval = -(sd(df_mex_case$log_wdi_overvalued)),
  time = 2,
  range = 10
)
```

```
## [1] "Error correction (EC) specified; dependent variable to be run in differences."
```

```
# View the results
summary(m_mex_flfp_mlfp)
```

```
##
## Call:
## lm(formula = as.formula(paste(paste(dvnamelist), "~", paste(colnames(IVs),
##     collapse = "+"), collapse = " ")))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1430 -0.4315  0.1366  0.4025  1.2742
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -169.0608    74.5714  -2.267  0.03860 *
## l.1.flfp_mlfp            -0.7693     0.2418  -3.182  0.00619 **
## ld.1.flfp_mlfp            0.1041     0.2299   0.453  0.65728
## ld.2.flfp_mlfp            0.3336     0.2125   1.570  0.13728
## d.1.log_wdi_overvalued   -5.9773     2.9751  -2.009  0.06287 .
## l.1.log_wdi_overvalued   -8.1464     2.6526  -3.071  0.00776 **
## d.1.log_gdpcap           11.1042    11.0243   1.007  0.32979
## l.1.log_gdpcap           23.1434     9.4791   2.442  0.02750 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7455 on 15 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.5538, Adjusted R-squared:  0.3456
## F-statistic:  2.66 on 7 and 15 DF,  p-value: 0.05296
```

```
# Table 4: Make the table ------------------------------------------------------

# Make variable names
list_var_names_mex_texreg <- list(
  'l.1.flfp'                =  'LDV$_{(t-1)}$',
  'ld.1.flfp'               =  '$\\Delta$LDV$_{(t-1)}$',
  'ld.2.flfp'               =  '$\\Delta$LDV$_{(t-2)}$',
  'l.1.flfp_mlfp'           =  'LDV$_{(t-1)}$',
  'ld.1.flfp_mlfp'          =  '$\\Delta$LDV$_{(t-1)}$',
  'ld.2.flfp_mlfp'          =  '$\\Delta$LDV$_{(t-2)}$',
  'l.1.log_wdi_overvalued'  =  'logOvervalued$_{(t-1)}$',
  'd.1.log_wdi_overvalued'  =  '$\\Delta$logOvervalued$_t$',
```

```
  'l.1.log_gdpcap'          =  'GDP$_{(t-1)}$',
  'd.1.log_gdpcap'          =  '$\\Delta$GDP$_t$',
  '(Intercept)'             = 'Constant'
)

# make the table
screenreg(
  list(m_mex_flfp,
       m_mex_flfp_mlfp),
  stars = c(0.01, 0.05, 0.1),
  # Coefficient labels
  custom.coef.map = list_var_names_mex_texreg,
  # Custom notes
  custom.note = "%stars",
)
```

```
##
## ================================================
##                             Model 1      Model 2
## ------------------------------------------------
## LDV$_{(t-1)}$                -0.82 ***    -0.77 ***
##                              (0.24)       (0.24)
## $\Delta$LDV$_{(t-1)}$         0.21         0.10
##                              (0.23)       (0.23)
## $\Delta$LDV$_{(t-2)}$         0.35         0.33
##                              (0.21)       (0.21)
## logOvervalued$_{(t-1)}$      -5.70 ***    -8.15 ***
##                              (1.80)       (2.65)
## $\Delta$logOvervalued$_t$    -3.25        -5.98 *
##                              (2.64)       (2.98)
## GDP$_{(t-1)}$                15.83 **     23.14 **
##                              (6.72)       (9.48)
## $\Delta$GDP$_t$               8.24        11.10
##                              (9.83)      (11.02)
## Constant                   -108.40 *    -169.06 **
##                             (52.30)      (74.57)
## ------------------------------------------------
## Num. obs.                    23.00        23.00
## ================================================
## *** p < 0.01; ** p < 0.05; * p < 0.1
```

```
######################################################################
##                 Table 5: Political Interest                      ##
######################################################################


# Table 5: Political Interest ---------------------------------------------

source("04_political_interest_results.R")
```

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest, : Some va
```

```
## `summarise()` has grouped output by 'cowcode'. You can override using the `.groups` argument.
```

```r
# View the table
tab_interest
```

```
##
## ============================================================
##                                          Model 1
## ------------------------------------------------------------
## Political Interest                        0.36 (0.45)
## Regime Type                               0.78 (0.34) **
## Political Interest $\times$ Regime Type  -1.94 (0.98) *
## GDP                                       0.05 (0.03) *
## Resource Rents                           -0.02 (0.01) **
## Exchange Rate: Narrow Crawling           -0.03 (0.07)
## Exchange Rate: Wide Crawling              0.06 (0.05)
## Exchange Rate: Freely Floating            0.06 (0.07)
## Exchange Rate: Freely Falling            -0.29 (0.05) ***
## Constant                                 -0.70 (0.23) ***
## ------------------------------------------------------------
## Num. obs.                                41
## ============================================================
## *** p < 0.01; ** p < 0.05; * p < 0.1,
##   Robust standard errors in parentheses
```

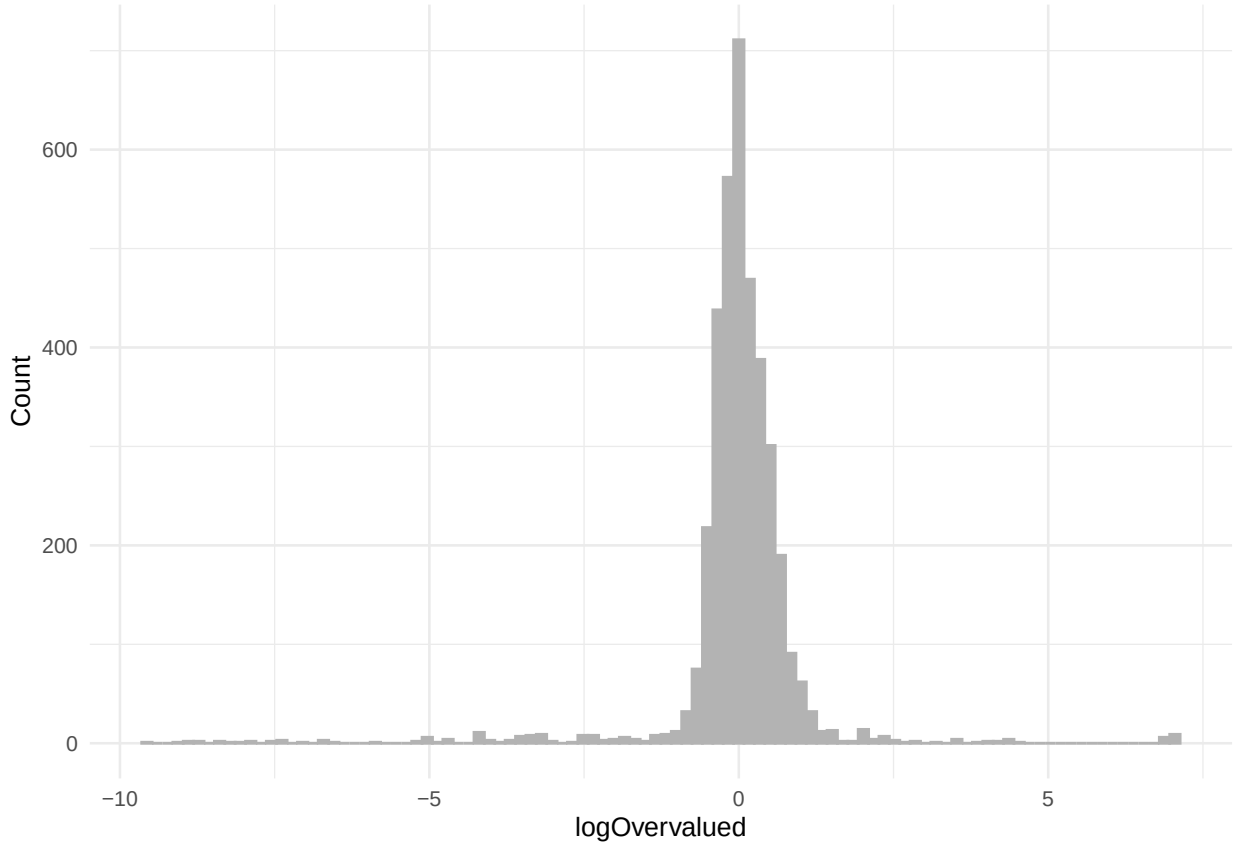
```r
######################################################################
##                            Figures                               ##
######################################################################

## Figure 1: Draw in Latex
# Figure 2: Overvalued Histogram ------------------------------------------

fig_hist_overvalued <-
  ggplot(data = df_tscs %>% filter(oecd == 0 &
                                     country != "Zimbabwe"),
         aes(x = log_wdi_overvalued)) +
  geom_histogram(bins = 100,
                 fill = "grey70",
                 color = "grey70") +
  labs(x = "logOvervalued", y = "Count")

fig_hist_overvalued
```

```
## Warning: Removed 606 rows containing non-finite values (`stat_bin()`).
```

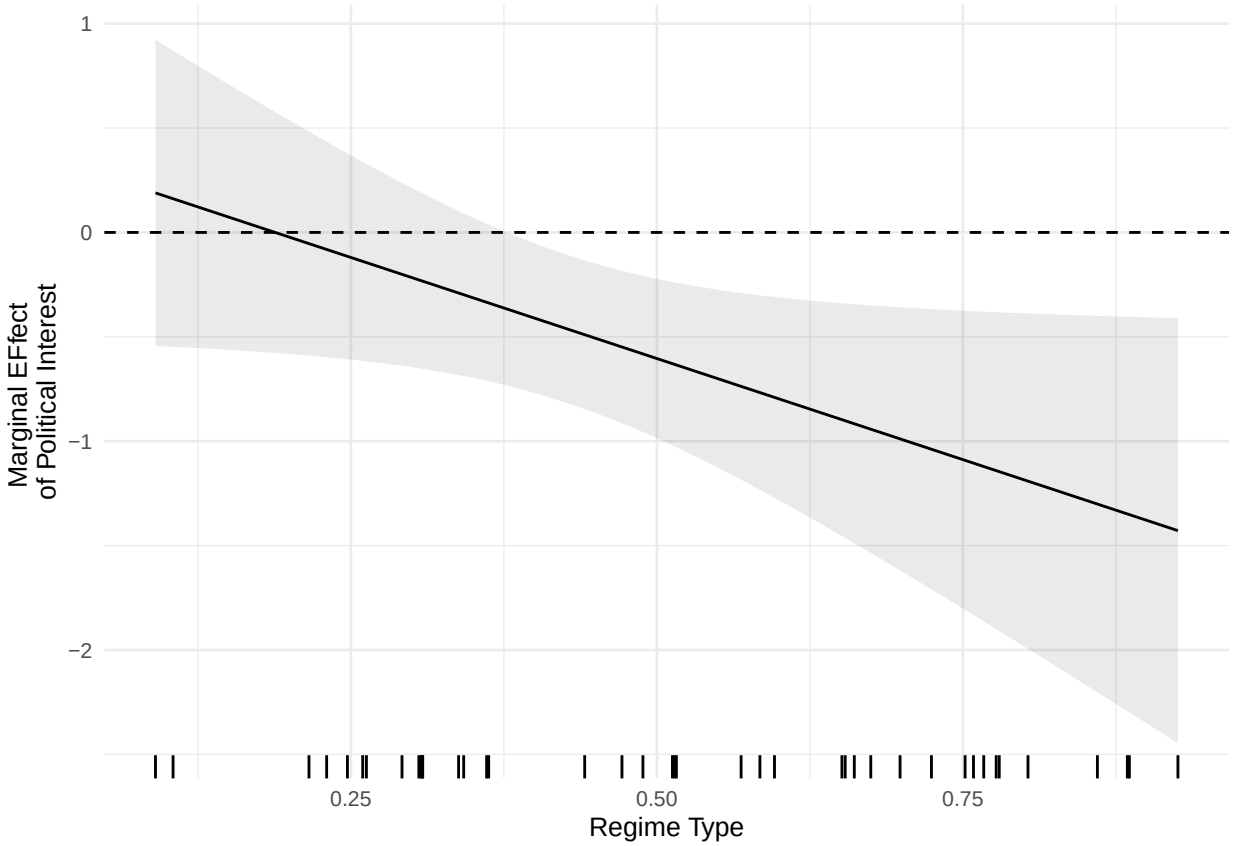



```
# Figure 3: Marginal Effect: logOvervalued ~ Interest * Regime Type -------

# Note, results require the `lm_interest` object created in the

# Make the plot
fig_me_interest_dem <-
  plot_slopes(
    lm_interest,
    variables = "some_interest_share",
    condition  = "vdem_elec_dem",
    vcov = "HC1",
    rug = TRUE
  ) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = "Regime Type",
       y = "Marginal EFfect \nof Political Interest")

fig_me_interest_dem
```




```
######################################################################
##                     Supplemental Appendix                        ##
######################################################################



# List of Countries ----------------------------------------------------

list_countries <- df_sum_stats %>%
  select(country) %>%
  group_by(country) %>%
  slice(1) %>%
  arrange(country) %>%
  ungroup()

# View the country list
data.frame(list_countries)
```

```
##                         country
## 1                   Afghanistan
## 2                       Albania
## 3                       Algeria
## 4                        Angola
## 5                     Argentina
## 6                       Armenia
## 7                    Azerbaijan
## 8                  Bahamas, The
```

```
## 9                        Bahrain
## 10                    Bangladesh
## 11                      Barbados
## 12                       Belarus
## 13                        Belize
## 14                         Benin
## 15                        Bhutan
## 16                       Bolivia
## 17         Bosnia and Herzegovina
## 18                      Botswana
## 19                        Brazil
## 20             Brunei Darussalam
## 21                      Bulgaria
## 22                  Burkina Faso
## 23                       Burundi
## 24                    Cabo Verde
## 25                      Cambodia
## 26                      Cameroon
## 27       Central African Republic
## 28                          Chad
## 29                         Chile
## 30                         China
## 31                      Colombia
## 32                       Comoros
## 33               Congo, Dem. Rep.
## 34                    Congo, Rep.
## 35                    Costa Rica
## 36                 Cote d'Ivoire
## 37                       Croatia
## 38                        Cyprus
## 39                Czech Republic
## 40            Dominican Republic
## 41                       Ecuador
## 42               Egypt, Arab Rep.
## 43                    El Salvador
## 44              Equatorial Guinea
## 45                       Eritrea
## 46                       Estonia
## 47                      Eswatini
## 48                      Ethiopia
## 49                          Fiji
## 50                         Gabon
## 51                   Gambia, The
## 52                       Georgia
## 53                         Ghana
## 54                     Guatemala
## 55                        Guinea
## 56                 Guinea-Bissau
## 57                        Guyana
## 58                         Haiti
## 59                      Honduras
## 60                       Hungary
## 61                         India
## 62                     Indonesia
```

```
## 63              Iran, Islamic Rep.
## 64                            Iraq
## 65                          Israel
## 66                         Jamaica
## 67                          Jordan
## 68                      Kazakhstan
## 69                           Kenya
## 70                     Korea, Rep.
## 71                          Kuwait
## 72                  Kyrgyz Republic
## 73                          Lao PDR
## 74                           Latvia
## 75                          Lebanon
## 76                          Lesotho
## 77                          Liberia
## 78                            Libya
## 79                        Lithuania
## 80                   Macedonia, FYR
## 81                       Madagascar
## 82                           Malawi
## 83                         Malaysia
## 84                         Maldives
## 85                             Mali
## 86                            Malta
## 87                       Mauritania
## 88                        Mauritius
## 89                           Mexico
## 90                          Moldova
## 91                         Mongolia
## 92                       Montenegro
## 93                          Morocco
## 94                       Mozambique
## 95                          Myanmar
## 96                          Namibia
## 97                            Nepal
## 98                        Nicaragua
## 99                            Niger
## 100                         Nigeria
## 101                            Oman
## 102                        Pakistan
## 103                          Panama
## 104               Papua New Guinea
## 105                        Paraguay
## 106                            Peru
## 107                     Philippines
## 108                          Poland
## 109                           Qatar
## 110                         Romania
## 111              Russian Federation
## 112                          Rwanda
## 113                           Samoa
## 114          Sao Tome and Principe
## 115                    Saudi Arabia
## 116                         Senegal
```

```
## 117                         Serbia
## 118                  Sierra Leone
## 119                     Singapore
## 120                Slovak Republic
## 121                      Slovenia
## 122                Solomon Islands
## 123                  South Africa
## 124                     Sri Lanka
## 125                     St. Lucia
## 126 St. Vincent and the Grenadines
## 127                         Sudan
## 128                      Suriname
## 129                    Tajikistan
## 130                      Tanzania
## 131                      Thailand
## 132                          Togo
## 133                         Tonga
## 134           Trinidad and Tobago
## 135                       Tunisia
## 136                        Turkey
## 137                  Turkmenistan
## 138                        Uganda
## 139                       Ukraine
## 140          United Arab Emirates
## 141                       Uruguay
## 142                    Uzbekistan
## 143                       Vanuatu
## 144                 Venezuela, RB
## 145                       Vietnam
## 146                   Yemen, Rep.
## 147                        Zambia
```

```
# LaTeX code was generated by hand

# SI: GMM Results  ------------------------------------------------------

cat(paste(
  "I conduct these tests in Stata using the following script:
  'scripts/03_stata_gmm_results.do'
  Use that script to replicate the results."))
```

```
## I conduct these tests in Stata using the following script:
##   'scripts/03_stata_gmm_results.do'
##   Use that script to replicate the results.
```

```
# Session Information ---------------------------------------------------

sessionInfo()
```

```
## R version 4.2.2 (2022-10-31)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Ventura 13.0
##
## Matrix products: default
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRlapack.dylib
##
```

```
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] here_1.0.1             kableExtra_1.3.4       modelsummary_1.4.0     texreg_1.38.6          marg
##  [7] bayestestR_0.13.1      fixest_0.11.1          dynamac_0.1.12         urca_1.3-3             spat
## [13] spdep_1.2-8            sf_1.0-12              spData_2.2.2           tscsdep_0.1.0          plm_
## [19] zoo_1.8-12             countrycode_1.4.0      broom_1.0.4            janitor_2.2.0          lubr
## [25] stringr_1.5.0          dplyr_1.1.2            purrr_1.0.1            readr_2.1.4            tidy
## [31] ggplot2_3.4.2          tidyverse_2.0.0        pacman_0.5.1
##
## loaded via a namespace (and not attached):
##   [1] uuid_1.1-0              backports_1.4.1         miscTools_0.6-28        systemfonts_1.0.4
##   [7] cshapes_2.0             digest_0.6.31           htmltools_0.5.5         fansi_1.0.4
##  [13] tzdb_0.4.0              vroom_1.6.3             officer_0.6.2           sandwich_3.0-2
##  [19] bdsmatrix_1.3-6         timechange_0.2.0        gfonts_0.2.0            colorspace_2.1-0
##  [25] rbibutils_2.2.13        xfun_0.39               crayon_1.5.2            jsonlite_1.8.4
##  [31] webshot_0.5.4           V8_4.3.0                maxLik_1.5-2            scales_1.2.1
##  [37] Rcpp_1.0.10             lfe_2.9-0               viridisLite_0.4.2       xtable_1.8-4
##  [43] proxy_0.4-27            Formula_1.2-5           DT_0.28                 fontLiberation_0.1.0
##  [49] httr_1.4.6              wk_0.7.3                ellipsis_0.3.2          farver_2.1.1
##  [55] deldir_1.0-6            utf8_1.2.3              crul_1.4.0              labeling_0.4.2
##  [61] later_1.3.1             munsell_0.5.0           tools_4.2.2             cachem_1.0.8
##  [67] evaluate_0.21           fastmap_1.1.1           ragg_1.2.5              yaml_2.3.7
##  [73] knitr_1.42              zip_2.3.0               s2_1.1.4                nlme_3.1-162
##  [79] rmapshaper_0.5.0        compiler_4.2.2          rstudioapi_0.14         curl_5.0.0
##  [85] gt_0.9.0                bslib_0.4.2             stringi_1.7.12          highr_0.10
##  [91] lattice_0.21-8          classInt_0.4-9          fontBitstreamVera_0.1.1 vctrs_0.6.2
##  [97] lifecycle_1.0.3         Rdpack_2.4              jquerylib_0.1.4         data.table_1.14.8
## [103] httpuv_1.6.11           R6_2.5.1                promises_1.2.0.1        KernSmooth_2.23-21
## [109] MASS_7.3-60             rprojroot_2.0.3         openssl_2.0.6           withr_2.5.0
## [115] parallel_4.2.2          hms_1.1.3               dreamerr_1.2.3          grid_4.2.2
## [121] rmarkdown_2.21          snakecase_0.11.0        numDeriv_2016.8-1.1     shiny_1.7.4
```