Component 2, Stage 1: DI -> SPI

Sevastian Sanchez

2025-09-02

```
knitr::opts_chunk$set(
  tidy = TRUE,
  tidy.opts = list(width.cutoff = 60)
)
```

Set up

```
# set working directory
setwd("~/Documents/GitHub/QMSS_Thesis_Sanchez")

# load libraries/packages
source("packages.R")
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## Loading required package: carData
##
##
## Attaching package: 'car'
##
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
##
## The following object is masked from 'package:purrr':
##
##     some
##
##
## Loading required package: usethis
##
##
```

```
## Attaching package: 'ERT'
##
##
## The following objects are masked from 'package:vdemdata':
##
##     codebook, vdem
##
##
##
## Please cite as:
##
##
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
##
##
##
## Attaching package: 'scales'
##
##
## The following object is masked from 'package:purrr':
##
##     discard
##
##
## The following object is masked from 'package:readr':
##
##     col_factor
##
##
##
## Attaching package: 'kableExtra'
##
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
##
##
##
## Attaching package: 'mice'
##
##
## The following object is masked from 'package:stats':
##
##     filter
##
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind
##
##
```

```
## Loading required package: MASS
##
##
## Attaching package: 'MASS'
##
##
## The following object is masked from 'package:dplyr':
##
##     select
##
##
##
## Attaching package: 'plm'
##
##
## The following objects are masked from 'package:dplyr':
##
##     between, lag, lead
##
##
##
## Attaching package: 'patchwork'
##
##
## The following object is masked from 'package:MASS':
##
##     area
##
##
##
## Attaching package: 'reshape2'
##
##
## The following object is masked from 'package:tidyr':
##
##     smiths
##
##
##
## Attaching package: 'jsonlite'
##
##
## The following object is masked from 'package:purrr':
##
##     flatten
##
##
## Loading required package: zoo
##
##
## Attaching package: 'zoo'
##
##
## The following objects are masked from 'package:base':
```

```
##
##     as.Date, as.Date.numeric
##
##
## Loading required package: Matrix
##
##
## Attaching package: 'Matrix'
##
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
##
## Loading required package: mvtnorm
##
## mediation: Causal Mediation Analysis
## Version: 4.5.0
##
##
##
## Attaching package: 'plotly'
##
##
## The following object is masked from 'package:MASS':
##
##     select
##
##
## The following object is masked from 'package:ggplot2':
##
##     last_plot
##
##
## The following object is masked from 'package:stats':
##
##     filter
##
##
## The following object is masked from 'package:graphics':
##
##     layout
##
##
##
## Attaching package: 'ggdag'
##
##
## The following object is masked from 'package:stats':
##
##     filter
```

```r
# load data
source("Comp2_panel_wrangling.R")
```

```
## Rows: 3340 Columns: 70
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (6): country_name, country_code, income_level, income_spi, region_spi, ...
## dbl (64): year, year_fct, sdg_overall, spi_comp, sci_overall, di_score, di_r...
##
## 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.
## Rows: 179 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): country_code, country_name
## dbl (4): in_merged_cleaned_spi, in_merged_cleaned_sdg, in_merged_exclusive, ...
##
## 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
# select path = 'data/Main CSV
# Outputs/merged_cleaned_sdg.csv'

panel_data_s1 <- panel_data %>%
    dplyr::select(country_name, country_code, year, spi_comp,
        di_score, di_score_lag1, di_score_lag2, log_gdppc, spi_comp_lag1,
        spi_comp_lag2, income_level_recoded) %>%
    dplyr::arrange(country_code, year)

head(panel_data_s1)
```

```
## # A tibble: 6 x 11
##   country_name country_code  year spi_comp di_score di_score_lag1 di_score_lag2
##   <chr>        <chr>        <dbl>    <dbl>    <dbl>         <dbl>         <dbl>
## 1 Afghanistan  AFG           2016     39.4     2.55          2.77          2.77
## 2 Afghanistan  AFG           2017     44.8     2.55          2.55          2.77
## 3 Afghanistan  AFG           2018     52.0     2.97          2.55          2.55
## 4 Afghanistan  AFG           2019     51.9     2.85          2.97          2.55
## 5 Afghanistan  AFG           2020     55.6     2.85          2.85          2.97
## 6 Afghanistan  AFG           2021     59.2     0.32          2.85          2.85
## # i 4 more variables: log_gdppc <dbl>, spi_comp_lag1 <dbl>,
## #   spi_comp_lag2 <dbl>, income_level_recoded <fct>
```

```r
# check lag structure is correct
head(panel_data_s1[, c("country_code", "year", "spi_comp", "spi_comp_lag1",
    "spi_comp_lag2", "di_score", "di_score_lag1", "di_score_lag2")])
```

```
## # A tibble: 6 x 8
##   country_code  year spi_comp spi_comp_lag1 spi_comp_lag2 di_score di_score_lag1
##   <chr>        <dbl>    <dbl>         <dbl>         <dbl>    <dbl>         <dbl>
## 1 AFG           2016     39.4          NA            NA        2.55          2.77
## 2 AFG           2017     44.8          39.4          NA        2.55          2.55
## 3 AFG           2018     52.0          44.8          39.4      2.97          2.55
## 4 AFG           2019     51.9          52.0          44.8      2.85          2.97
## 5 AFG           2020     55.6          51.9          52.0      2.85          2.85
```

```
## 6 AFG           2021     59.2           55.6          51.9     0.32          2.85
## # i 1 more variable: di_score_lag2 <dbl>
```

```r
dim(panel_data_s1)  # 1336 rows, 11 columns
```

```
## [1] 1336   11
```

Stage 1 Models:

ols_spi_di ols_spi_di_L1 ols_spi_di_L2

fd_spi_di fd_spi_di_L1 fd_spi_di_L2

fe_spi_di fe_spi_di_L1 fe_spi_di_L2

1.1) POLS [Stage 1]: SPI ~ DI

The effect of democracy on SPI Performance

```r
# Contemporaneous Effect: SPI ~ DI
ols_spi_di <- plm(formula = spi_comp ~ di_score + log_gdppc +
    factor(income_level_recoded) + factor(year), index = c("country_code",
    "year"), model = "pooling", data = panel_data_s1)
summary(ols_spi_di, vcov = vcovHC(ols_spi_di, cluster = "group",
    type = "HC1"))
```

```
## Pooling Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovHC(ols_spi_di, cluster = "group", type =
##
## Call:
## plm(formula = spi_comp ~ di_score + log_gdppc + factor(income_level_recoded) +
##     factor(year), data = panel_data_s1, model = "pooling", index = c("country_code",
##     "year"))
##
## Unbalanced Panel: n = 156, T = 2-8, N = 1236
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -33.45307  -5.37204   0.81769   6.43191  27.34450
##
## Coefficients:
##                                Estimate Std. Error t-value  Pr(>|t|)
## (Intercept)                    15.23814    9.84001  1.5486   0.12174
## di_score                        3.57776    0.55129  6.4897 1.246e-10 ***
## log_gdppc                       2.70897    1.61721  1.6751   0.09417 .
## factor(income_level_recoded)1   1.27699    2.59806  0.4915   0.62315
## factor(income_level_recoded)2   1.85970    4.20090  0.4427   0.65807
## factor(income_level_recoded)3   3.26998    6.36952  0.5134   0.60778
## factor(year)2017                2.44631    0.25432  9.6191 < 2.2e-16 ***
## factor(year)2018                4.71508    0.39505 11.9355 < 2.2e-16 ***
## factor(year)2019                5.07356    0.44979 11.2798 < 2.2e-16 ***
## factor(year)2020                7.85757    0.53232 14.7610 < 2.2e-16 ***
## factor(year)2021               12.68678    0.65200 19.4581 < 2.2e-16 ***
## factor(year)2022               11.88282    0.68832 17.2635 < 2.2e-16 ***
```

```
## factor(year)2023               13.43830    0.74322 18.0811 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    331520
## Residual Sum of Squares: 132770
## R-Squared:      0.5995
## Adj. R-Squared: 0.59557
## F-statistic: 120.55 on 12 and 155 DF, p-value: < 2.22e-16
```

```
# Adding Lag1: SPI ~ DI
ols_spi_di_L1 <- plm(formula = spi_comp ~ di_score + di_score_lag1 +
    log_gdppc + factor(income_level_recoded) + factor(year),
    index = c("country_code", "year"), model = "pooling", data = panel_data_s1)
summary(ols_spi_di_L1, vcov = vcovHC(ols_spi_di_L1, cluster = "group",
    type = "HC1"))
```

```
## Pooling Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovHC(ols_spi_di_L1, cluster = "group", type
##
## Call:
## plm(formula = spi_comp ~ di_score + di_score_lag1 + log_gdppc +
##     factor(income_level_recoded) + factor(year), data = panel_data_s1,
##     model = "pooling", index = c("country_code", "year"))
##
## Unbalanced Panel: n = 156, T = 2-8, N = 1236
##
## Residuals:
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -33.4261  -5.4299   0.8220   6.3808  26.8326
##
## Coefficients:
##                               Estimate Std. Error t-value Pr(>|t|)
## (Intercept)                   14.53708    9.78922  1.4850  0.13780
## di_score                      -0.25841    1.52845 -0.1691  0.86577
## di_score_lag1                  3.89794    1.48176  2.6306  0.00863 **
## log_gdppc                      2.75956    1.60877  1.7153  0.08654 .
## factor(income_level_recoded)1  1.02740    2.57016  0.3997  0.68942
## factor(income_level_recoded)2  1.64231    4.19227  0.3917  0.69531
## factor(income_level_recoded)3  3.01125    6.34121  0.4749  0.63496
## factor(year)2017               2.40403    0.25354  9.4818  < 2e-16 ***
## factor(year)2018               4.87608    0.40562 12.0212  < 2e-16 ***
## factor(year)2019               5.08222    0.44442 11.4357  < 2e-16 ***
## factor(year)2020               7.69425    0.52907 14.5430  < 2e-16 ***
## factor(year)2021              12.46236    0.63517 19.6205  < 2e-16 ***
## factor(year)2022              12.06651    0.69467 17.3702  < 2e-16 ***
## factor(year)2023              13.30608    0.72808 18.2756  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    331520
## Residual Sum of Squares: 131860
## R-Squared:      0.60225
## Adj. R-Squared: 0.59802
```

```
## F-statistic: 110.68 on 13 and 155 DF, p-value: < 2.22e-16
```

```r
# Adding Lag2: SPI ~ DI
ols_spi_di_L2 <- plm(formula = spi_comp ~ di_score + di_score_lag1 +
    di_score_lag2 + log_gdppc + factor(income_level_recoded) +
    factor(year), index = c("country_code", "year"), model = "pooling",
    data = panel_data_s1)
summary(ols_spi_di_L2, vcov = vcovHC(ols_spi_di_L2, cluster = "group",
    type = "HC1"))
```

```
## Pooling Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovHC(ols_spi_di_L2, cluster = "group", type
##
## Call:
## plm(formula = spi_comp ~ di_score + di_score_lag1 + di_score_lag2 +
##     log_gdppc + factor(income_level_recoded) + factor(year),
##     data = panel_data_s1, model = "pooling", index = c("country_code",
##         "year"))
##
## Unbalanced Panel: n = 156, T = 2-8, N = 1236
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -32.96813  -5.30461   0.77702   6.43025  27.02016
##
## Coefficients:
##                               Estimate Std. Error t-value  Pr(>|t|)
## (Intercept)                   13.42990    9.76938  1.3747 0.1694787
## di_score                       0.14694    1.44862  0.1014 0.9192221
## di_score_lag1                 -1.17616    0.81639 -1.4407 0.1499295
## di_score_lag2                  4.74434    1.40726  3.3713 0.0007715 ***
## log_gdppc                      2.90011    1.60302  1.8092 0.0706724 .
## factor(income_level_recoded)1  0.61137    2.55351  0.2394 0.8108176
## factor(income_level_recoded)2  1.17829    4.19069  0.2812 0.7786294
## factor(income_level_recoded)3  2.27224    6.32999  0.3590 0.7196838
## factor(year)2017               2.19042    0.27248  8.0389 2.125e-15 ***
## factor(year)2018               4.59684    0.40109 11.4608 < 2.2e-16 ***
## factor(year)2019               5.07591    0.44480 11.4117 < 2.2e-16 ***
## factor(year)2020               7.52619    0.52689 14.2843 < 2.2e-16 ***
## factor(year)2021              12.06758    0.64906 18.5924 < 2.2e-16 ***
## factor(year)2022              11.56918    0.66530 17.3894 < 2.2e-16 ***
## factor(year)2023              13.34045    0.72316 18.4475 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    331520
## Residual Sum of Squares: 130520
## R-Squared:      0.6063
## Adj. R-Squared: 0.60179
## F-statistic: 99.5282 on 14 and 155 DF, p-value: < 2.22e-16
```

POLS Summary Table

1.2) First Difference [Stage 1]: SPI ~ DI

```
# Contemporaneous Effect: SPI ~ DI
fd_spi_di <- plm(formula = spi_comp ~ di_score + log_gdppc +
    factor(income_level_recoded), index = c("country_code", "year"),
    data = fd_data, model = "fd")
summary(fd_spi_di, vcov = vcovHC(fd_spi_di, cluster = "group",
    type = "HC1"))
```

```
## Oneway (individual) effect First-Difference Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovHC(fd_spi_di, cluster = "group", type = "H
##
## Call:
## plm(formula = spi_comp ~ di_score + log_gdppc + factor(income_level_recoded),
##     data = fd_data, model = "fd", index = c("country_code", "year"))
##
## Unbalanced Panel: n = 156, T = 2-8, N = 1236
## Observations used in estimation: 1080
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -10.76953  -1.97929  -0.52327   1.78058  14.84939
##
## Coefficients:
##                                Estimate Std. Error t-value  Pr(>|t|)
## (Intercept)                    1.768828   0.093338 18.9507 < 2.2e-16 ***
## di_score                      -0.742541   0.342914 -2.1654  0.030578 *
## log_gdppc                      2.733410   0.862117  3.1706  0.001564 **
## factor(income_level_recoded)1 -0.171106   0.914686 -0.1871  0.851645
## factor(income_level_recoded)2 -1.030236   1.134502 -0.9081  0.364032
## factor(income_level_recoded)3 -2.495450   1.341701 -1.8599  0.063171 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    11247
## Residual Sum of Squares: 11097
## R-Squared:      0.01336
## Adj. R-Squared: 0.008767
## F-statistic: 3.95079 on 5 and 155 DF, p-value: 0.0021209
```

```
# Adding Lag1: SPI ~ DI
fd_spi_di_L1 <- plm(formula = spi_comp ~ di_score + di_score_lag1 +
    log_gdppc + factor(income_level_recoded), index = c("country_code",
    "year"), data = fd_data, model = "fd")
summary(fd_spi_di_L1, vcov = vcovHC(fd_spi_di_L1, cluster = "group",
    type = "HC1"))
```

```
## Oneway (individual) effect First-Difference Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovHC(fd_spi_di_L1, cluster = "group", type =
##
## Call:
```

```
## plm(formula = spi_comp ~ di_score + di_score_lag1 + log_gdppc +
##     factor(income_level_recoded), data = fd_data, model = "fd",
##     index = c("country_code", "year"))
##
## Unbalanced Panel: n = 156, T = 2-8, N = 1236
## Observations used in estimation: 1080
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -10.98115  -1.98153  -0.54502   1.72205  14.69291
##
## Coefficients:
##                                Estimate Std. Error t-value  Pr(>|t|)
## (Intercept)                    1.752412   0.093445 18.7533 < 2.2e-16 ***
## di_score                      -0.692968   0.356597 -1.9433  0.052244 .
## di_score_lag1                 -0.542485   0.371272 -1.4612  0.144266
## log_gdppc                      2.724490   0.864013  3.1533  0.001659 **
## factor(income_level_recoded)1 -0.142872   0.926988 -0.1541  0.877540
## factor(income_level_recoded)2 -0.979737   1.143868 -0.8565  0.391905
## factor(income_level_recoded)3 -2.483881   1.351851 -1.8374  0.066428 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    11247
## Residual Sum of Squares: 11081
## R-Squared:      0.014807
## Adj. R-Squared: 0.0092978
## F-statistic: 4.01444 on 6 and 155 DF, p-value: 0.00091368
```

```r
# Adding Lag2: SPI ~ DI
fd_spi_di_L2 <- plm(formula = spi_comp ~ di_score + di_score_lag1 +
    di_score_lag2 + log_gdppc + factor(income_level_recoded),
    index = c("country_code", "year"), data = fd_data, model = "fd")
summary(fd_spi_di_L2, vcov = vcovHC(fd_spi_di_L2, cluster = "group",
    type = "HC1"))
```

```
## Oneway (individual) effect First-Difference Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovHC(fd_spi_di_L2, cluster = "group", type =
##
## Call:
## plm(formula = spi_comp ~ di_score + di_score_lag1 + di_score_lag2 +
##     log_gdppc + factor(income_level_recoded), data = fd_data,
##     model = "fd", index = c("country_code", "year"))
##
## Unbalanced Panel: n = 156, T = 2-8, N = 1236
## Observations used in estimation: 1080
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -10.84975  -1.98361  -0.56395   1.68412  15.44794
##
## Coefficients:
##                                Estimate Std. Error t-value  Pr(>|t|)
## (Intercept)                    1.783983   0.093183 19.1450 < 2.2e-16 ***
```

```
## di_score                        -0.753782   0.360994 -2.0881  0.037027 *
## di_score_lag1                   -0.654580   0.380920 -1.7184  0.086009 .
## di_score_lag2                    1.157114   0.428697  2.6991  0.007061 **
## log_gdppc                        2.808883   0.873183  3.2168  0.001335 **
## factor(income_level_recoded)1 -0.377437   0.943192 -0.4002  0.689111
## factor(income_level_recoded)2 -1.180075   1.163561 -1.0142  0.310720
## factor(income_level_recoded)3 -2.723653   1.363021 -1.9982  0.045942 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    11247
## Residual Sum of Squares: 11006
## R-Squared:      0.021435
## Adj. R-Squared: 0.015045
## F-statistic: 4.02648 on 7 and 155 DF, p-value: 0.00044398
```

First Difference Summary Table

1.3) Fixed Effects [Stage 1]: SPI ~ DI

```r
# Contemporaneous Effect: SPI ~ DI
fe_spi_di <- plm(formula = spi_comp ~ di_score + log_gdppc +
    factor(income_level_recoded) + factor(year), index = c("country_code",
    "year"), data = panel_data_s1, model = "within")
summary(fe_spi_di, vcov = vcovHC(fe_spi_di, cluster = "group",
    type = "HC1"))
```

```
## Oneway (individual) effect Within Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovHC(fe_spi_di, cluster = "group", type = "
##
## Call:
## plm(formula = spi_comp ~ di_score + log_gdppc + factor(income_level_recoded) +
##     factor(year), data = panel_data_s1, model = "within", index = c("country_code",
##     "year"))
##
## Unbalanced Panel: n = 156, T = 2-8, N = 1236
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -15.60516  -2.02061   0.03228   2.03210  15.01236
##
## Coefficients:
##                                 Estimate Std. Error t-value Pr(>|t|)
## di_score                        -0.036223   0.558021 -0.0649   0.9483
## log_gdppc                        0.683486   1.713533  0.3989   0.6901
## factor(income_level_recoded)1  1.198321   2.045044  0.5860   0.5580
## factor(income_level_recoded)2 -0.245138   2.327339 -0.1053   0.9161
## factor(income_level_recoded)3  0.649600   2.489953  0.2609   0.7942
## factor(year)2017                 2.565264   0.260316  9.8544   <2e-16 ***
## factor(year)2018                 5.009475   0.438123 11.4339   <2e-16 ***
## factor(year)2019                 5.254521   0.485121 10.8314   <2e-16 ***
## factor(year)2020                 7.600472   0.532321 14.2780   <2e-16 ***
## factor(year)2021                12.333688   0.706079 17.4679   <2e-16 ***
```

```
## factor(year)2022              11.757715   0.767884 15.3118   <2e-16 ***
## factor(year)2023              12.966012   0.846059 15.3252   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    38485
## Residual Sum of Squares: 12750
## R-Squared:      0.66871
## Adj. R-Squared: 0.6169
## F-statistic: 62.4033 on 12 and 155 DF, p-value: < 2.22e-16
```

```r
# Adding Lag1: SPI ~ DI
fe_spi_di_L1 <- plm(formula = spi_comp ~ di_score + di_score_lag1 +
    log_gdppc + factor(income_level_recoded) + factor(year),
    index = c("country_code", "year"), data = panel_data_s1,
    model = "within")
summary(fe_spi_di_L1, vcov = vcovHC(fe_spi_di_L1, cluster = "group",
    type = "HC1"))
```

```
## Oneway (individual) effect Within Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovHC(fe_spi_di_L1, cluster = "group", type =
##
## Call:
## plm(formula = spi_comp ~ di_score + di_score_lag1 + log_gdppc +
##     factor(income_level_recoded) + factor(year), data = panel_data_s1,
##     model = "within", index = c("country_code", "year"))
##
## Unbalanced Panel: n = 156, T = 2-8, N = 1236
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -15.59358  -2.01471   0.05272   2.04189  14.99043
##
## Coefficients:
##                                Estimate Std. Error t-value Pr(>|t|)
## di_score                      -0.32632    0.57729 -0.5653   0.5720
## di_score_lag1                  0.39768    0.53722  0.7403   0.4593
## log_gdppc                      0.68538    1.71276  0.4002   0.6891
## factor(income_level_recoded)1  1.16458    2.04364  0.5699   0.5689
## factor(income_level_recoded)2 -0.28233    2.32981 -0.1212   0.9036
## factor(income_level_recoded)3  0.63264    2.48941  0.2541   0.7994
## factor(year)2017               2.56442    0.26050  9.8444   <2e-16 ***
## factor(year)2018               5.02873    0.43887 11.4584   <2e-16 ***
## factor(year)2019               5.26092    0.48591 10.8269   <2e-16 ***
## factor(year)2020               7.59732    0.53219 14.2755   <2e-16 ***
## factor(year)2021              12.33353    0.70515 17.4906   <2e-16 ***
## factor(year)2022              11.79808    0.77282 15.2662   <2e-16 ***
## factor(year)2023              12.98117    0.84696 15.3268   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    38485
## Residual Sum of Squares: 12743
## R-Squared:      0.66888
```

```
## Adj. R-Squared: 0.61675
## F-statistic: 58.0614 on 13 and 155 DF, p-value: < 2.22e-16
```

```r
# Adding Lag2: SPI ~ DI
fe_spi_di_L2 <- plm(formula = spi_comp ~ di_score + di_score_lag1 +
    di_score_lag2 + log_gdppc + factor(income_level_recoded) +
    factor(year), index = c("country_code", "year"), data = panel_data_s1,
    model = "within")
summary(fe_spi_di_L2, vcov = vcovHC(fe_spi_di_L2, cluster = "group",
    type = "HC1"))
```

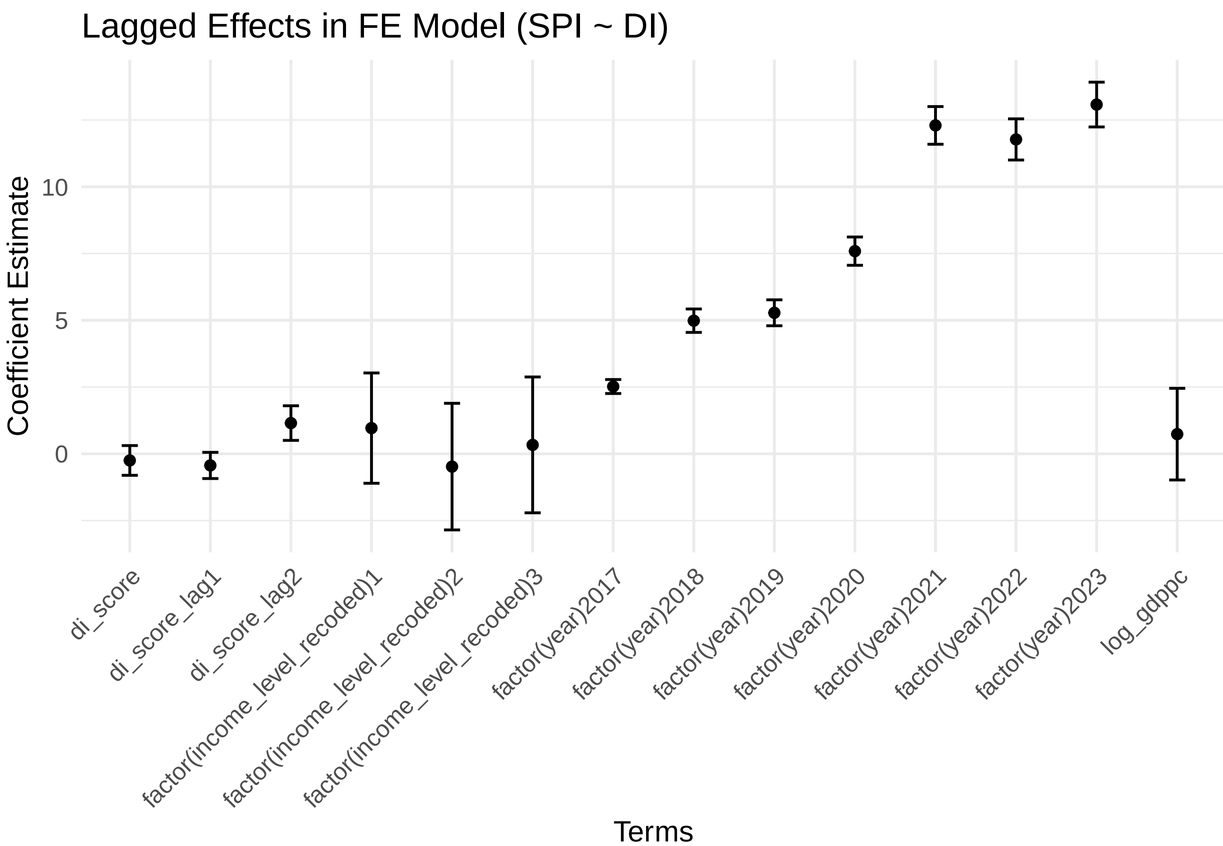
```
## Oneway (individual) effect Within Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovHC(fe_spi_di_L2, cluster = "group", type =
##
## Call:
## plm(formula = spi_comp ~ di_score + di_score_lag1 + di_score_lag2 +
##     log_gdppc + factor(income_level_recoded) + factor(year),
##     data = panel_data_s1, model = "within", index = c("country_code",
##         "year"))
##
## Unbalanced Panel: n = 156, T = 2-8, N = 1236
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -16.103326  -1.992448   0.038386   2.056200  14.890068
##
## Coefficients:
##                               Estimate Std. Error t-value Pr(>|t|)
## di_score                      -0.24729    0.55706 -0.4439  0.65719
## di_score_lag1                 -0.43464    0.49046 -0.8862  0.37571
## di_score_lag2                  1.15091    0.64795  1.7762  0.07598 .
## log_gdppc                      0.73730    1.71636  0.4296  0.66759
## factor(income_level_recoded)1  0.96323    2.06498  0.4665  0.64098
## factor(income_level_recoded)2 -0.47917    2.37096 -0.2021  0.83988
## factor(income_level_recoded)3  0.33444    2.54343  0.1315  0.89541
## factor(year)2017               2.52061    0.26195  9.6223  < 2e-16 ***
## factor(year)2018               4.98539    0.43888 11.3594  < 2e-16 ***
## factor(year)2019               5.28050    0.48630 10.8586  < 2e-16 ***
## factor(year)2020               7.58975    0.52991 14.3228  < 2e-16 ***
## factor(year)2021              12.29724    0.70617 17.4139  < 2e-16 ***
## factor(year)2022              11.77297    0.77053 15.2790  < 2e-16 ***
## factor(year)2023              13.07786    0.83829 15.6007  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    38485
## Residual Sum of Squares: 12690
## R-Squared:      0.67026
## Adj. R-Squared: 0.61799
## F-statistic: 53.1239 on 14 and 155 DF, p-value: < 2.22e-16
```

Fixed Effects Summary Table

stargazer table for only lag2 models

Fixed Effects Error Bar Visualization

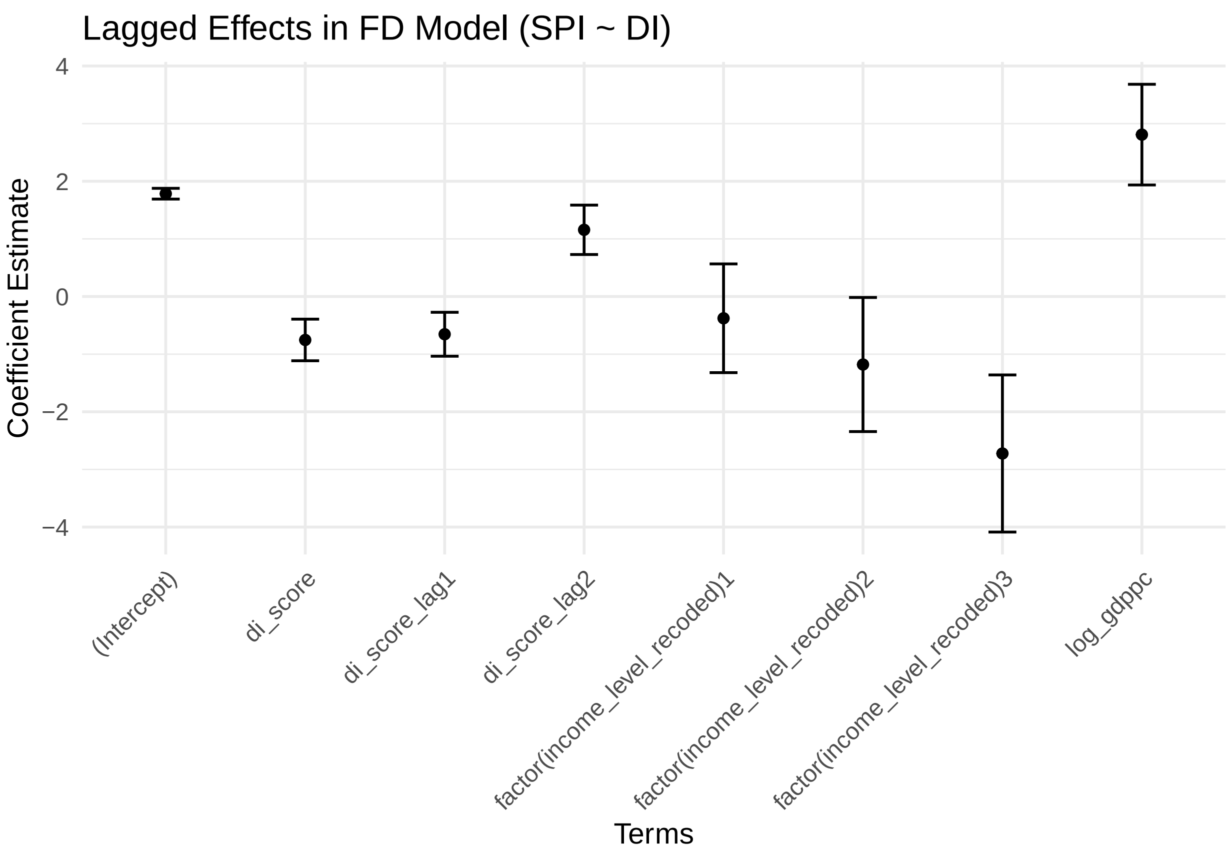
```
# Extract coefficients and robust standard errors from the
# FE model
fe_spi_di_L2_results <- summary(fe_spi_di_L2, vcov = vcovHC(fe_spi_di_L2,
    cluster = "group", type = "HC1"))
# Create a data frame for visualization
coef_df <- data.frame(term = rownames(fe_spi_di_L2_results$coefficients),
    estimate = fe_spi_di_L2_results$coefficients[, "Estimate"],
    std.error = fe_spi_di_L2_results$coefficients[, "Std. Error"])
# Create a ggplot error bar chart
ggplot(coef_df, aes(x = term, y = estimate)) + geom_point() +
    geom_errorbar(aes(ymin = estimate - std.error, ymax = estimate +
        std.error), width = 0.2) + labs(title = "Lagged Effects in FE Model (SPI ~ DI)",
    x = "Terms", y = "Coefficient Estimate") + theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# Save the plot
# ggsave('figures/error_bar_fe_spi_di_L2.png', width = 10,
# height = 6)
```

First Difference Error Bar Visualization

```r
# Extract coefficients and robust standard errors from the
# FD model
fd_spi_di_L2_results <- summary(fd_spi_di_L2, vcov = vcovHC(fd_spi_di_L2,
    cluster = "group", type = "HC1"))
# Create a data frame for visualization
coef_fd_df <- data.frame(term = rownames(fd_spi_di_L2_results$coefficients),
    estimate = fd_spi_di_L2_results$coefficients[, "Estimate"],
    std.error = fd_spi_di_L2_results$coefficients[, "Std. Error"])
# Create a ggplot error bar chart
ggplot(coef_fd_df, aes(x = term, y = estimate)) + geom_point() +
    geom_errorbar(aes(ymin = estimate - std.error, ymax = estimate +
        std.error), width = 0.2) + labs(title = "Lagged Effects in FD Model (SPI ~ DI)",
    x = "Terms", y = "Coefficient Estimate") + theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
# Save the plot

# ggsave('figures/error_bar_fd_spi_di_L2', width = 10,
# height = 6)
```

Check for Autocorrelation

```r
# APPLY Wooldridge Test for AR(1) Errors in FE Panel
# Models: pwartest()
# https://search.r-project.org/CRAN/refmans/plm/html/pwartest.html
```

```r
# This is MUCH BETTER for panel data with small T AND
# unbalanced panels!!!
pwartest(fe_spi_di_L2)  # [significant]
```

```
##
##  Wooldridge's test for serial correlation in FE panels
##
## data:  fe_spi_di_L2
## F = 498.66, df1 = 1, df2 = 1078, p-value < 2.2e-16
## alternative hypothesis: serial correlation
```

Significant p-value indicates the presence of autocorrelation in the residuals of the fixed effects model. This suggests that the errors are correlated over time, which violates one of the key assumptions of linear regression models.

This is corrected by using robust standard errors clustered by country, which accounts for the potential autocorrelation in the residuals.

Check for Heteroskedasticity

```r
# Apply Breusch-Pagan test for heteroskedasticity
bptest(fe_spi_di_L2, studentize = TRUE)  # Heteroskedasticity [significant]
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fe_spi_di_L2
## BP = 139.49, df = 14, p-value < 2.2e-16
```

```r
bptest(fd_spi_di_L2, studentize = TRUE)  # Heteroskedasticity [significant]
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fd_spi_di_L2
## BP = 133.14, df = 7, p-value < 2.2e-16
```

The Breusch-Pagan test indicates the presence of heteroskedasticity in the residuals of the fixed effects model. This suggests that the variance of the errors is not constant across observations, which violates another key assumption of linear regression models.

##[IGNORE ALL BELOW] ## MODEL MISSPECIFICATION CHECKS [Stage 1] ========================== Following standard econometric practice, I first establish the appropriate functional form for main effects, using RESET tests and residual diagnostics, before testing for theoretically motivated interaction effects.

RESET Test for Misspecification [STAGE 1]

The RESET test checks for omitted variable bias by testing if the squared and cubed fitted values significantly improve the model fit. If the p-value is low (typically < 0.05), it suggests that the model may be misspecified, indicating that a non-linear relationship or omitted variables are present.

When you get a p-value of less than 0.05 this means that you reject the null hypothesis - this means that statistically your model is better with first order terms.

We reject the Null Hypothesis of no misspecification in the relationship between DI and SPI performance, as the p-value is greater than 0.05 in in the FE model. This suggests that non-linear terms may make a significant contribution to the model.

Stepwise Check: Applying Polynomial Terms [Stage 1]

For this section, all continuous predictors (DI & Log(GDP)) are centered for simplicity and to avoid multicollinearity issues.

- **H0:** The relationship between SPI and DI performance is linear.
- **H1:** The relationship between SPI and DI performance is non-linear (quadratic or cubic).

DI: We fail to reject the null hypothesis of linearity in the relationship between DI and SPI performance, as the p-value is greater than 0.05 in all models. Although literature suggests a non-linear relationship (SOURCE), the results here do not support it.

MODEL DIAGNOSTICS [Stage 1]

Multicollinearity Assessment and Correction [Stage 2]

[CHECK BACK W GREG - NEED TO FIGURE OUT WHAT TO DO WITH THIS]

The only model worth looking for multicollinearity is the base model, as the other models have no polynomial or interaction terms.