First Tests: OLS, Interactions, Subgroups

Sevastian Sanchez

2025-04-21

All Countries, Preliminary Analysis (SPI x SDGs)

#FIRST: Libraries, Directory & Data

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

#load libraries
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: '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
```

```r
#load data
source("data/data_sources.R")
```

```
## Rows: 4340 Columns: 80
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): country, iso3c, income, region
## dbl (76): date, SPI.INDEX.PIL1, SPI.INDEX.PIL2, SPI.INDEX.PIL3, SPI.INDEX.PI...
##
## 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: 2618 Columns: 32
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (31): country_name, country_code, IQ.SCI.OVRL, IQ.SCI.MTHD, IQ.SCI.PRDC,...
## dbl  (1): Year
##
## 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: 17024 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

```
## chr (2): country_name, country_code
## dbl (2): year, gdp_pc
##
## 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: 17350 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): country_name
## dbl (16): year, ccodecow, country_id, infcap_pca, infcap_irt, statagency, ce...
##
## 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: 8288 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): country_code, income_level
## dbl (1): year
##
## 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: 2958 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): country_name, country_code
## dbl (2): year, di_score
##
## 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.
```

#SECOND: run function in r-script: df_years_function.R

```
#df_years: used for extracting data from specified year to now

#load function
source("df_years()_Function.R")
```

```
## Rows: 4340 Columns: 80
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): country, iso3c, income, region
## dbl (76): date, SPI.INDEX.PIL1, SPI.INDEX.PIL2, SPI.INDEX.PIL3, SPI.INDEX.PI...
##
## 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: 2618 Columns: 32
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (31): country_name, country_code, IQ.SCI.OVRL, IQ.SCI.MTHD, IQ.SCI.PRDC,...
## dbl  (1): Year
##
## 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: 17024 Columns: 4
## -- Column specification --------------------------------------------------------
```

```
## Delimiter: ","
## chr (2): country_name, country_code
## dbl (2): year, gdp_pc
##
## 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: 17350 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): country_name
## dbl (16): year, ccodecow, country_id, infcap_pca, infcap_irt, statagency, ce...
##
## 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: 8288 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): country_code, income_level
## dbl (1): year
##
## 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: 2958 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): country_name, country_code
## dbl (2): year, di_score
##
## 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.
```

```
#specify start year
merged <- df_years(yr1 = 2000)
```

```
## Warning: There were 4 warnings in `dplyr::mutate()`.
## The first warning was:
## i In argument: `across(...)`.
## Caused by warning:
## ! NAs introduced by coercion
## i Run `dplyr::last_dplyr_warnings()` to see the 3 remaining warnings.
```

##Vdem wrangling / cleaning (skip if df_years() used)

SPI wrangling / cleaning (skip if df_years() used)

SCI wrangling / cleaning (skip if df_years() used)

##SDG wrangling / cleaning (skip if df_years() used)

##GDP_PC wrangling / cleaning (skip if df_years() used)

Merging & subsetting (skip if df_years() used)

All countries 2019 data only

Set up & missing data

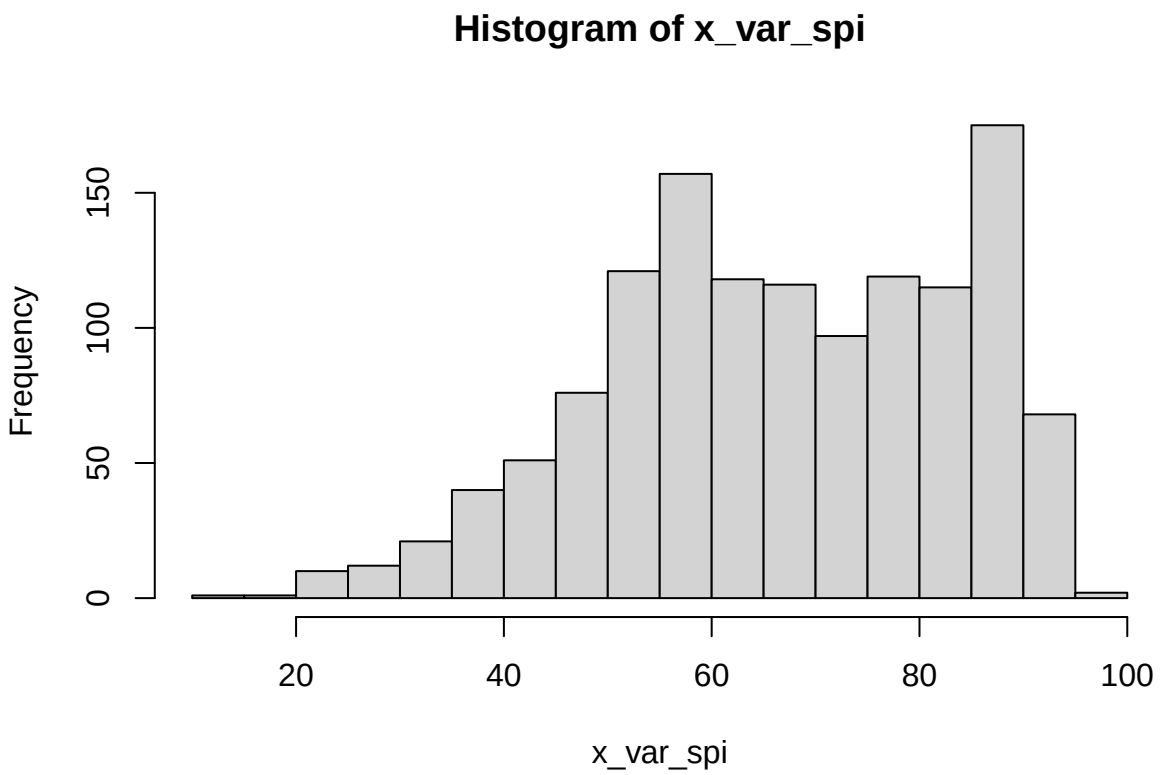
```
# x & y variables
y_var_sdg <- merged$sdg_overall
x_var_spi <- merged$spi_comp
x_var_sci <- merged$sci_overall
df_sdg_statcap <- data.frame(y_var_sdg, x_var_spi, x_var_sci)

# how many na's?
colSums(is.na(df_sdg_statcap))
```

```
## y_var_sdg x_var_spi x_var_sci
##         0      3068      2313
```

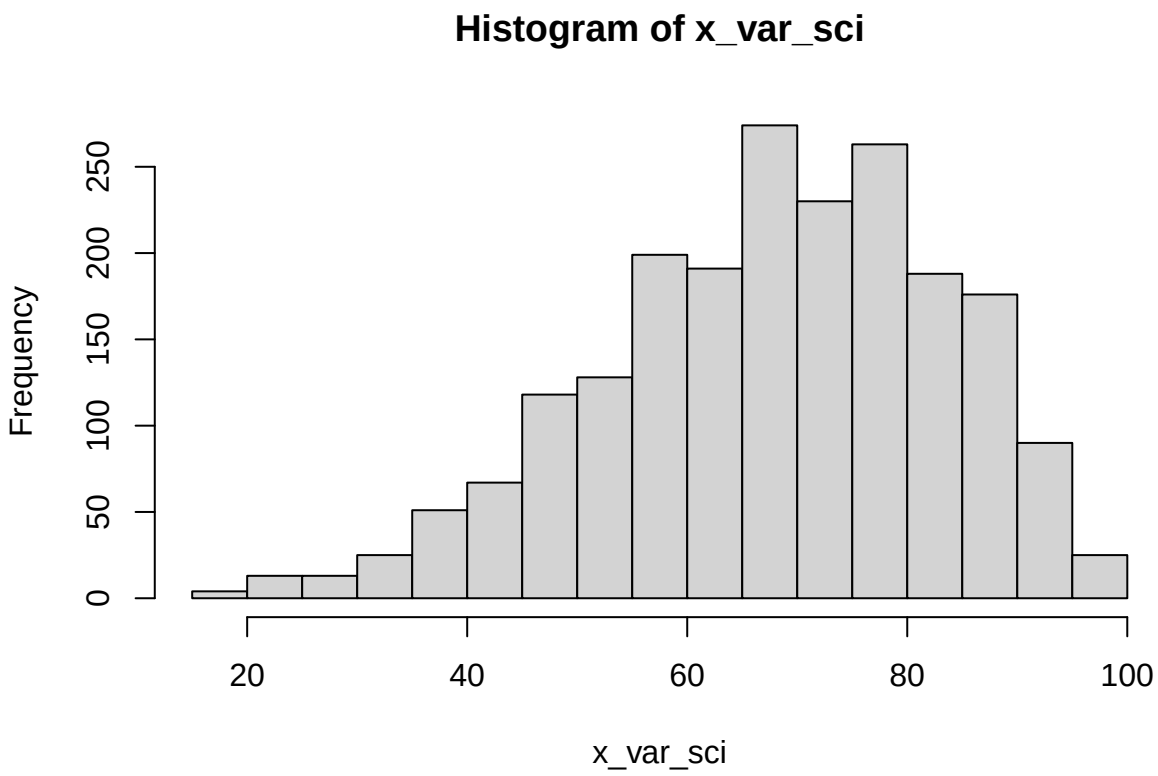
```
#how many observations
colSums(!is.na(df_sdg_statcap))
```

```
## y_var_sdg x_var_spi x_var_sci
##      4368      1300      2055
```

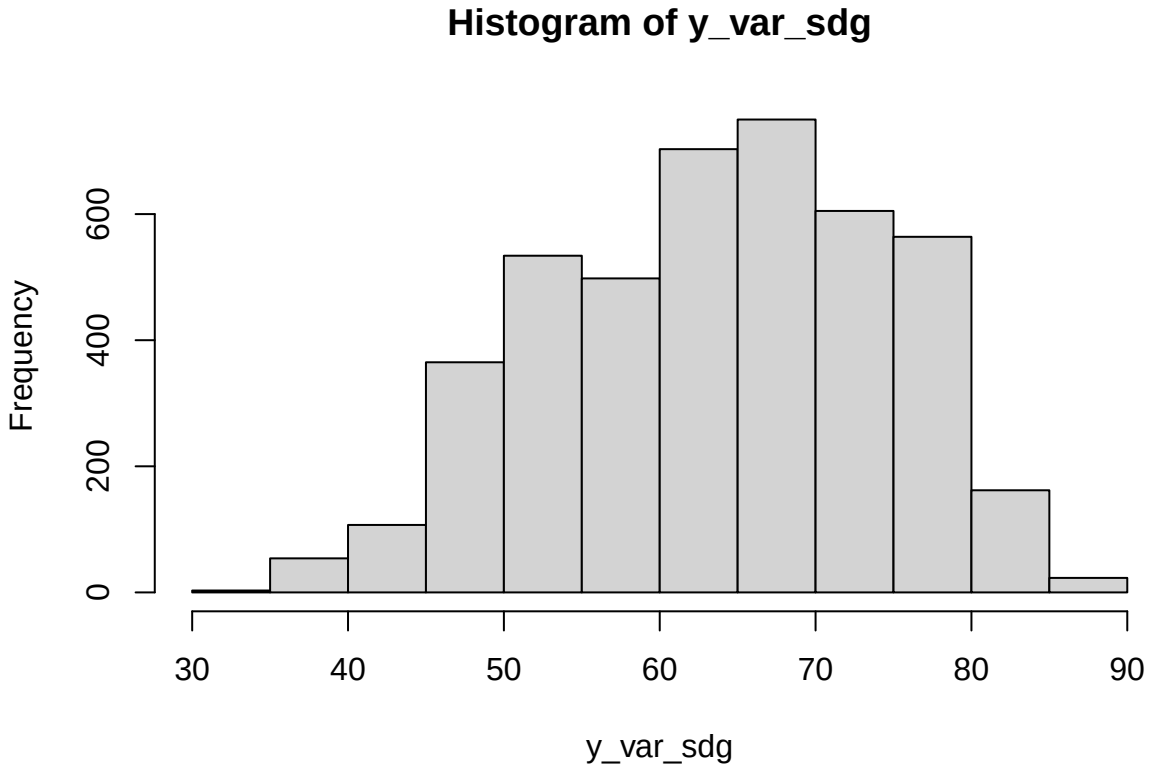
```
hist(x_var_spi)
```




```
hist(x_var_sci)
```




```
hist(y_var_sdg)
```



Number of observations in SPI: 1300; number of NA's 3068 (not all years captured) Number of observations in SCI: 2055; number of NA's 2313 (not all countries captured)

COMPARING SPI & SCI X VARIABLES

Aggregated SPI & SDG Scores
H0: Null, there is no relationship
H1: there is a statistically significant relationship between overall SPI and SDG composite scores

```
#correlation coefficients (r-squared), WITHOUT control variables

#x-var 1 = spi
correlation_sdg_spi <- cor(y_var_sdg, x_var_spi, use = "complete.obs")^2

#x-var 2 = sci
correlation_sdg_sci <- cor(y_var_sdg, x_var_sci, use = "complete.obs")^2

# pasting result
string_corcoef <- "Correlation coefficient:"
paste(string_corcoef, correlation_sdg_spi, "(SPI)", correlation_sdg_sci, "(SCI)")
```

```
## [1] "Correlation coefficient: 0.616037202309322 (SPI) 0.417965563339242 (SCI)"
```

Correlation coefficient/R-sq (SPI): 0.616037202309322
Correlation coefficient/R-sq (SCI): 0.417965563339242

Comparing SPI & SCI to identify best model, w/o controls

Finding estimated impact of variables on SDG status prior to adding controls

```
ols_spi_naive <- lm(y_var_sdg ~ x_var_spi)
summary(ols_spi_naive)
```

```
##
## Call:
## lm(formula = y_var_sdg ~ x_var_spi)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.3175  -4.4186   0.5969   4.4301  20.1684
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.94626    0.72064   48.49   <2e-16 ***
## x_var_spi    0.47806    0.01048   45.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.338 on 1298 degrees of freedom
##   (3068 observations deleted due to missingness)
## Multiple R-squared:  0.616,  Adjusted R-squared:  0.6157
## F-statistic:  2083 on 1 and 1298 DF,  p-value: < 2.2e-16
```

```
# 2. OLS for SCI and SDG - Overall
ols_sci_naive <- lm(y_var_sdg ~ x_var_sci)
summary(ols_sci_naive)
```

```
##
## Call:
## lm(formula = y_var_sdg ~ x_var_sci)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.0240  -4.9307   0.2307   4.8361  18.8180
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.88189    0.71150   47.62   <2e-16 ***
## x_var_sci    0.39209    0.01021   38.40   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.166 on 2053 degrees of freedom
##   (2313 observations deleted due to missingness)
## Multiple R-squared:  0.418,  Adjusted R-squared:  0.4177
## F-statistic:  1474 on 1 and 2053 DF,  p-value: < 2.2e-16
```

```
# 3. Multiple Regression with both SPI and SCI
ols_multiple_naive <- lm(y_var_sdg ~ x_var_spi + x_var_sci)
summary(ols_multiple_naive)
```

```
##
## Call:
## lm(formula = y_var_sdg ~ x_var_spi + x_var_sci)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.5483  -5.4484   0.4037   4.7941  17.9050
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.86438    1.27744  28.075  < 2e-16 ***
## x_var_spi    0.28779    0.03369   8.542  < 2e-16 ***
## x_var_sci    0.15311    0.03232   4.738  2.7e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.671 on 593 degrees of freedom
##   (3772 observations deleted due to missingness)
## Multiple R-squared:  0.4651, Adjusted R-squared:  0.4633
## F-statistic: 257.8 on 2 and 593 DF,  p-value: < 2.2e-16
```

ols_spi_naive: 0.47806 (p-value < 0.001)
ols_sci_naive: 0.39209 (p-value < 0.001)
ols_multiple_naive: spi: 0.28779 (p-value < 0.001); sci: 0.15311 (p-value < 0.001)

The impact of SCI on SDG and SPI on SDG are statistically significant, in all models. SPI appears to have a greater impact on SDGs compared to that of SCI, regardless of the model. All of this is wothout controls.

Comparing SPI & SCI to identify best model, WITH controls

H0: Null, SCI model > SPI model
H1: SPI model > SCI model

```
# 1. OLS for SPI and SDG - Overall
ols_spi <- lm(y_var_sdg ~ x_var_spi + log_gdppc + population, data = merged)
```

```r
summary(ols_spi)
```

```
##
## Call:
## lm(formula = y_var_sdg ~ x_var_spi + log_gdppc + population,
##     data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.4288  -3.1699   0.0989   3.2398  11.8636
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.753e+01  8.874e-01  19.750   <2e-16 ***
## x_var_spi    2.800e-01  1.152e-02  24.312   <2e-16 ***
## log_gdppc    3.563e+00  1.309e-01  27.218   <2e-16 ***
## population  -1.359e-09  9.176e-10  -1.482    0.139
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.862 on 1100 degrees of freedom
##   (3264 observations deleted due to missingness)
## Multiple R-squared:  0.7702, Adjusted R-squared:  0.7696
## F-statistic:  1229 on 3 and 1100 DF,  p-value: < 2.2e-16
```

```r
# 2. OLS for SCI and SDG - Overall
ols_sci <- lm(y_var_sdg ~ x_var_sci + log_gdppc + population, data = merged)
summary(ols_sci)
```

```
##
## Call:
## lm(formula = y_var_sdg ~ x_var_sci + log_gdppc + population,
##     data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.7451  -3.2153   0.0107   3.2418  15.8327
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.567e+00  8.110e-01   4.399 1.14e-05 ***
## x_var_sci    2.268e-01  7.955e-03  28.512  < 2e-16 ***
## log_gdppc    5.365e+00  1.145e-01  46.840  < 2e-16 ***
## population  -1.813e-09  6.460e-10  -2.806  0.00506 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.926 on 2021 degrees of freedom
##   (2343 observations deleted due to missingness)
## Multiple R-squared:  0.724,  Adjusted R-squared:  0.7235
## F-statistic:  1767 on 3 and 2021 DF,  p-value: < 2.2e-16
```

```r
# 3. Multiple Regression with both SPI and SCI
ols_multiple <- lm(y_var_sdg ~ x_var_spi + x_var_sci + log_gdppc + population, data = merged)
summary(ols_multiple)
```

```
##
## Call:
## lm(formula = y_var_sdg ~ x_var_spi + x_var_sci + log_gdppc +
##     population, data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3413  -2.7930  -0.0514   2.5643  13.5455
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.478e+00  1.471e+00   2.364   0.0184 *
## x_var_spi    1.216e-01  2.305e-02   5.277 1.86e-07 ***
## x_var_sci    1.288e-01  2.148e-02   5.994 3.61e-09 ***
## log_gdppc    5.528e+00  2.010e-01  27.502  < 2e-16 ***
## population  -2.882e-09  1.005e-09  -2.866   0.0043 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.345 on 576 degrees of freedom
##   (3787 observations deleted due to missingness)
## Multiple R-squared:  0.764,  Adjusted R-squared:  0.7624
## F-statistic: 466.2 on 4 and 576 DF,  p-value: < 2.2e-16
```

We reject the null hypothesis that there is no relationship between SPI and SDG composite scores. Holding all else constant (log gdp per capita and population), there is a positive moderate relationship between Statistical Performance (SPI) and SDG status.

ols_spi: 0.280 (p-value < 0.001)
ols_sci: 0.268 (p-value < 0.001)
ols_multiple: spi: 0.1216 (p-value < 0.001); sci: 0.1288 (p-value < 0.001)

Comparing coefficients, SPI has a greater impact on SDG status (0.280) than SCI (0.268). However, in a multiple regression model containing both SPI and SCI, SCI has more of an impact on SDG status (0.1288) (net of SPI) than SPI (0.1216) (net of SCI). Here the coefficients represent the unique impact of each x variable on SDG status, net of all other variables.

Model 1 (ols_spi) does not control for SCI and model 2 (ols_sci) does not control for spi – this is okay. SPI is the predecessor of the SCI, sharing/data overlap, and so it is expected to have significant statistical correlation (multicollinearity). This is likely what explains the significant reduction of coefficients (from 0.280 to 0.1216 for SPI, and from 0.268 to 0.1288 for SCI) indicating that they're both capturing much of the same underlying relationship with SDG status.

AIC/BIC Checking Fit [FIX TEST- #N DIFFERS BTW MODELS]

```
# Compare all three models with AIC
#AIC(ols_spi, ols_sci, ols_multiple)

# Compare all three models with BIC
#BIC(ols_spi, ols_sci, ols_multiple)
```

Best fit: ols_spi (Adj Rsq: 0.7696) (AIC/BIC: _____)

SPI & SCI colinearity VIF

```
# Check correlation between SPI and SCI
cor(x_var_spi, x_var_sci, use = "complete.obs")
```

```
## [1] 0.8276634
```

```
# Check VIF (Variance Inflation Factor)
vif(ols_multiple)
```

```
##  x_var_spi  x_var_sci  log_gdppc population
##   3.346118   3.206861   1.266844   1.023637
```

colinearity: there is significant co-linearity between SCI and SPI, with a correlation of 0.8277. Upon integrating within the same model, SCI inflated the standard error of SPI from 0.0115 to 0.0231. SCI had a similar reaction fro the SPI with its standard error increasing from 0.00796 to 0.0215.

VIF: Such multicolinearity is reflected by the VIF test which accounts for all x variables in the model instead of just the two measures of statistical capacity (SCI & SPI).

x_var_spi: 3.34 x_var_sci: 3.21 log_gdppc: 1.27 population: 1.02

Unsurprisingly, the variance of the SPI coefficient is inflated by a factor of 3.34 due to correlation with other predictors. Similarly, the SCI coefficient's variance is inflated by 3.21 times. However, it is acceptable to use in the same model as it will not severely impact estimates given both factors are less than 5.0.

###Combine to single index: principle component analysis (FOR FUTURE)

```
# Standardize both measures
#spi_z <- scale(x_var_spi)
#sci_z <- scale(x_var_sci)

# Create composite (simple average)
#stat_capacity_index <- (spi_z + sci_z)/2

#extract common variance
#pca_result <- prcomp(cbind(x_var_spi, x_var_sci), scale = TRUE)
#stat_capacity_pc1 <- pca_result$x[,1]  # First principal component
```

Selecting model: My research question is about overall statistical capacity rather than comparing different measures. The SPI model reveals a better fit than the SCI model (Adj Rsq: 0.7696 > 0.7235). It is also a slightly better fit compared to the multiple OLS model containing both SPI and SCI (Adj Rsq: 0.7696 > 0.7624).

Visual Analysis: SCI & SPI x SDG

```
#visualize differences in fit
library(ggplot2)
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

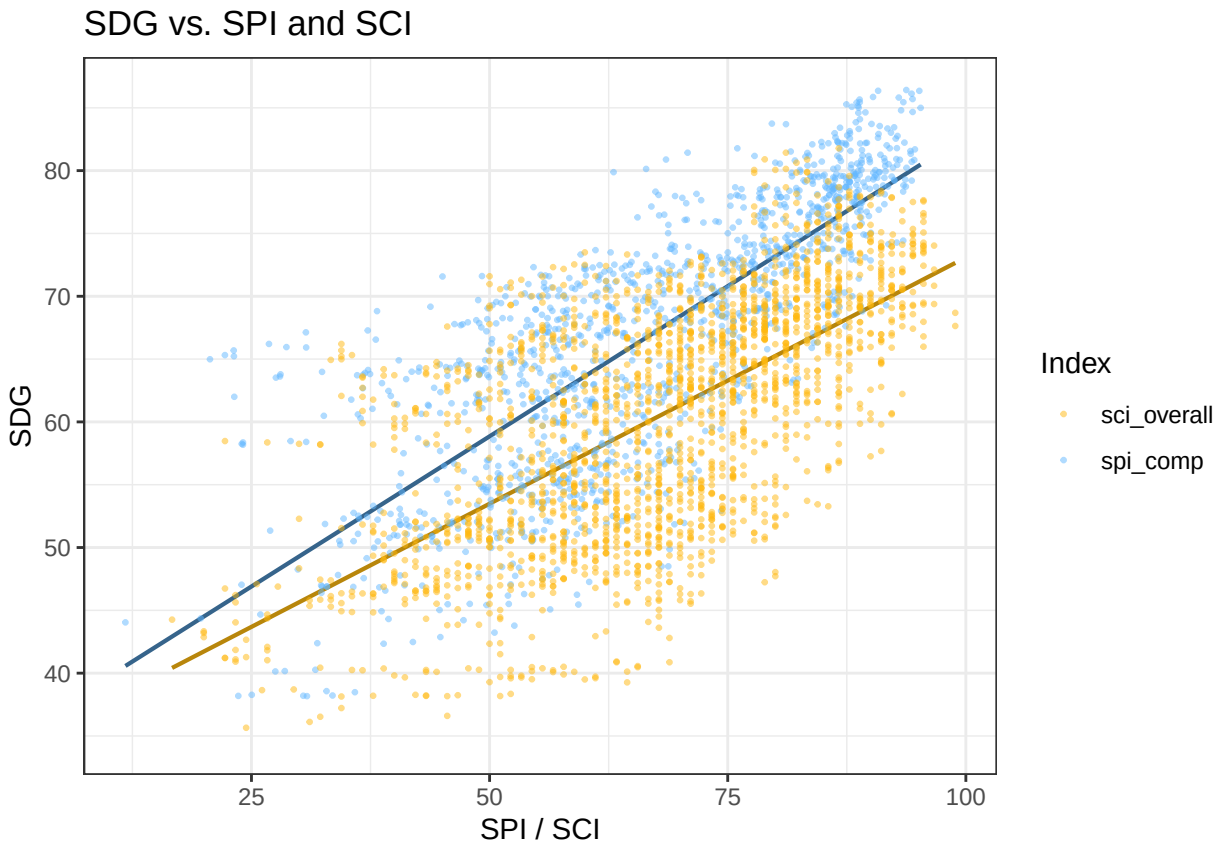
```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## 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
```

```
#define regression line colors
spi_line <- "steelblue4"
sci_line <- "darkgoldenrod"

# Creating scatterplot with both SPI and SCI on the same plot
Compare_fit <- ggplot(merged, aes(x = spi_comp, y = sdg_overall))+
  geom_smooth(aes(x = spi_comp, y = sdg_overall),
              color = spi_line,
              method = "lm",
              linewidth = 0.75,
              se = FALSE)+ # Regression line SPI
  geom_smooth(aes(x = sci_overall, y = sdg_overall),
              color = sci_line,
              method = "lm",
              linewidth = 0.75,
              se = FALSE)+ # Regression line for SCI
  geom_point(aes(color = "spi_comp"), alpha=0.50, size = 0.5)+ # Scatter plot for SPI
  geom_point(aes(x = sci_overall, y = sdg_overall, color = "sci_overall"),
             alpha=0.5, size = 0.5)+ # Add SCI points w/different color
  scale_color_manual(values = c("spi_comp" = "steelblue1",
                                "sci_overall" = "darkgoldenrod1")) +
  labs(title = "SDG vs. SPI and SCI",
       x = "SPI / SCI",
       y = "SDG",
       color = "Index") +  # Title for legend
  theme_bw() # Optional: adds a clean, black and white theme

Compare_fit
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 3068 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 2313 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 3068 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 2313 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
#make interactive
#ggplotly(Compare_fit)

# Save to specific folder
# ggsave("~/Documents/GitHub/QMSS_Thesis_Sanchez/Output_CSVs/fd_plot.png", p, width = 10, height = 6)
```

INTERACTIONS AND SUBGROUP ANALYSIS

##Checking for Interactions: - Is there a need for subgroup analysis, and if so, by what kind of group? - Options: GNI Classification (income_level), regime_type_2, regime_type_4, di_score

```
#interaction: does GNI Classification (income_level) affect the relationship between x (spi) & y (sdg)?
inc_lev_interaction <- lm(sdg_overall ~ spi_comp + spi_comp*income_level + log_gdppc + population,
                          data = merged)
summary(inc_lev_interaction)
```

```
##
## Call:
## lm(formula = sdg_overall ~ spi_comp + spi_comp * income_level +
##     log_gdppc + population, data = merged)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.648  -2.608   0.080   2.486  13.487
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)              2.512e+01  3.161e+00   7.946 4.77e-15 ***
## spi_comp                 4.393e-01  1.812e-02  24.236  < 2e-16 ***
## income_levelL            8.567e+00  2.242e+00   3.822 0.000140 ***
## income_levelLM           3.234e+00  2.037e+00   1.587 0.112752
## income_levelUM           2.165e+01  1.852e+00  11.695  < 2e-16 ***
## log_gdppc                1.529e+00  2.944e-01   5.194 2.46e-07 ***
## population              -2.940e-09  8.069e-10  -3.644 0.000281 ***
## spi_comp:income_levelL  -2.630e-01  3.308e-02  -7.949 4.65e-15 ***
## spi_comp:income_levelLM -6.954e-02  2.700e-02  -2.576 0.010138 *
## spi_comp:income_levelUM -2.920e-01  2.418e-02 -12.077  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.205 on 1094 degrees of freedom
##   (3264 observations deleted due to missingness)
## Multiple R-squared:  0.8291, Adjusted R-squared:  0.8277
## F-statistic: 589.6 on 9 and 1094 DF,  p-value: < 2.2e-16
```

```
#interaction: does regime_type_2 affect the relationship between x (spi) & y (sdg)?
reg_type2_interaction <- lm(sdg_overall ~ spi_comp + spi_comp*regime_type_2 + log_gdppc + population,
                      data = merged)
summary(reg_type2_interaction)
```

```
##
## Call:
## lm(formula = sdg_overall ~ spi_comp + spi_comp * regime_type_2 +
##     log_gdppc + population, data = merged)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.674  -3.181   0.091   3.195  12.390
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.742e+01  1.249e+00  13.941   <2e-16 ***
## spi_comp                  2.766e-01  1.516e-02  18.247   <2e-16 ***
## regime_type_21            3.348e+00  1.404e+00   2.385   0.0172 *
## log_gdppc                 3.526e+00  1.378e-01  25.578   <2e-16 ***
## population               -8.581e-10  9.207e-10  -0.932   0.3516
## spi_comp:regime_type_21 -2.968e-02  2.116e-02  -1.403   0.1609
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.824 on 1098 degrees of freedom
##   (3264 observations deleted due to missingness)
## Multiple R-squared:  0.7742, Adjusted R-squared:  0.7732
## F-statistic: 752.9 on 5 and 1098 DF,  p-value: < 2.2e-16
```

```
#interaction: does regime_type_4 affect the relationship between x (spi) & y (sdg)?
reg_type4_interaction <- lm(sdg_overall ~ spi_comp + spi_comp*regime_type_4 + log_gdppc + population,
                      data = merged)
summary(reg_type4_interaction)
```

```
##
## Call:
```

```
## lm(formula = sdg_overall ~ spi_comp + spi_comp * regime_type_4 +
##     log_gdppc + population, data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.7949  -3.1091  -0.0311   3.2177  12.3414
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.751e+01  1.923e+00   9.106  < 2e-16 ***
## spi_comp                 2.220e-01  2.956e-02   7.509 1.23e-13 ***
## regime_type_41          -1.949e+00  1.920e+00  -1.015   0.3103
## regime_type_42           1.164e+00  2.110e+00   0.552   0.5811
## regime_type_43           1.614e+00  3.152e+00   0.512   0.6087
## log_gdppc                3.760e+00  1.572e-01  23.916  < 2e-16 ***
## population              -6.644e-10  9.222e-10  -0.720   0.4714
## spi_comp:regime_type_41  6.015e-02  3.365e-02   1.787   0.0742 .
## spi_comp:regime_type_42  2.915e-02  3.488e-02   0.836   0.4035
## spi_comp:regime_type_43  1.136e-02  4.330e-02   0.262   0.7931
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.805 on 1094 degrees of freedom
##   (3264 observations deleted due to missingness)
## Multiple R-squared:  0.7768, Adjusted R-squared:  0.7749
## F-statistic: 422.9 on 9 and 1094 DF,  p-value: < 2.2e-16
```

```
#interaction: does di affect the relationship between x (spi) & y (sdg)?
reg_type_di_interaction <- lm(sdg_overall ~ spi_comp + spi_comp*di_score + log_gdppc + population,
                      data = merged)
summary(reg_type_di_interaction)
```

```
##
## Call:
## lm(formula = sdg_overall ~ spi_comp + spi_comp * di_score + log_gdppc +
##     population, data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.4455  -3.3270  -0.0258   3.1958  12.6393
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.660e+01  1.966e+00   8.440   <2e-16 ***
## spi_comp             2.907e-01  2.509e-02  11.588   <2e-16 ***
## di_score             6.905e-01  3.072e-01   2.247   0.0248 *
## log_gdppc            3.391e+00  1.520e-01  22.306   <2e-16 ***
## population          -1.150e-09  9.170e-10  -1.255   0.2099
## spi_comp:di_score -5.504e-03  4.440e-03  -1.240   0.2154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.846 on 1077 degrees of freedom
##   (3285 observations deleted due to missingness)
## Multiple R-squared:   0.77,  Adjusted R-squared:  0.769
```

```
## F-statistic: 721.2 on 5 and 1077 DF,  p-value: < 2.2e-16
```

GNI Income Classification: Yes there are statistically significant interactions found from GNI classifications that affects the relationship between spi and sdgs

Binary Regime Type: No there is no statistically significant interactions found from regime type (autocracy vs democracy) that affects the relationship between spi and sdgs.

Categorical Regime type (4 options): No there is no statistically significant interactions found from regime type (Closed autocracy, electoral autocracy, electoral democracy, liberal democracy) that affects the relationship between spi and sdgs.

Continuous di_score [0-1] Regime type: No there is no statistically significant interactions found from regime type (infinate between 0-1) that affects the relationship between spi and sdgs.

WB GNI Classifications: income_level ("H", "UM", "LM", "L")

Disaggregated/Grouped by Development Status: Make 4 regression models and then put them all together in a table to compare the slopes and R-sq values.

```
# 1. Overall model (all countries)
overall_lm <- lm(sdg_overall ~ spi_comp + di_score + log_gdppc + population + gini,
                         data = merged)
summary(overall_lm)
```

```
##
## Call:
## lm(formula = sdg_overall ~ spi_comp + di_score + log_gdppc +
##     population + gini, data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.2634  -2.5312   0.5357   2.6313   8.1247
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.463e+01  2.118e+00  16.353   <2e-16 ***
## spi_comp     2.474e-01  2.010e-02  12.309   <2e-16 ***
## di_score     6.733e-02  1.445e-01   0.466   0.6415
## log_gdppc    2.991e+00  2.262e-01  13.223   <2e-16 ***
## population  -1.767e-09  7.891e-10  -2.240   0.0256 *
## gini        -2.450e+01  2.696e+00  -9.086   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.723 on 463 degrees of freedom
##   (3899 observations deleted due to missingness)
## Multiple R-squared:  0.7921, Adjusted R-squared:  0.7898
## F-statistic: 352.7 on 5 and 463 DF,  p-value: < 2.2e-16
```

```
# 2. High income countries
high_inc_lm <- lm(sdg_overall ~ spi_comp + di_score + log_gdppc + population + gini,
                         data = merged %>%
                           filter(income_level == "H"))
summary(high_inc_lm)
```

```
##
## Call:
```

```
## lm(formula = sdg_overall ~ spi_comp + di_score + log_gdppc +
##     population + gini, data = merged %>% filter(income_level ==
##     "H"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5021 -1.5521 -0.0159  1.5926  5.0537
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.431e+01  4.449e+00  18.949  < 2e-16 ***
## spi_comp     1.887e-01  2.541e-02   7.424 3.15e-12 ***
## di_score     1.937e+00  2.248e-01   8.617 2.02e-15 ***
## log_gdppc   -2.493e+00  3.791e-01  -6.576 4.08e-10 ***
## population  -1.156e-10  3.192e-09  -0.036    0.971
## gini        -3.396e+01  3.801e+00  -8.936 2.61e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.231 on 201 degrees of freedom
##   (873 observations deleted due to missingness)
## Multiple R-squared:  0.6637, Adjusted R-squared:  0.6553
## F-statistic: 79.33 on 5 and 201 DF,  p-value: < 2.2e-16
```

```r
# 3. Upper-middle income countries
upper_mid_lm <- lm(sdg_overall ~ spi_comp + di_score + log_gdppc + population + gini,
                   data = merged %>%
                     filter(income_level == "UM"))
summary(upper_mid_lm)
```

```
##
## Call:
## lm(formula = sdg_overall ~ spi_comp + di_score + log_gdppc +
##     population + gini, data = merged %>% filter(income_level ==
##     "UM"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1207 -1.1353 -0.0782  0.8816  5.3862
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.040e+01  5.205e+00  11.605  < 2e-16 ***
## spi_comp     1.202e-01  1.881e-02   6.390 2.30e-09 ***
## di_score     6.881e-01  1.604e-01   4.290 3.31e-05 ***
## log_gdppc    1.166e+00  6.195e-01   1.882   0.0619 .
## population   6.995e-10  7.825e-10   0.894   0.3729
## gini        -3.013e+01  3.057e+00  -9.855  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.139 on 140 degrees of freedom
##   (758 observations deleted due to missingness)
## Multiple R-squared:  0.5393, Adjusted R-squared:  0.5228
## F-statistic: 32.78 on 5 and 140 DF,  p-value: < 2.2e-16
```

```r
# 4. Lower-middle income countries
lower_mid_lm <- lm(sdg_overall ~ spi_comp + di_score + log_gdppc + population + gini,
                        data = merged %>%
                          filter(income_level == "LM"))
summary(lower_mid_lm)
```

```
##
## Call:
## lm(formula = sdg_overall ~ spi_comp + di_score + log_gdppc +
##     population + gini, data = merged %>% filter(income_level ==
##     "LM"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0352 -1.8603  0.4498  2.8106  6.4091
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.731e+01  8.527e+00   3.202  0.00194 **
## spi_comp     2.666e-01  5.126e-02   5.201 1.43e-06 ***
## di_score    -3.808e-01  3.338e-01  -1.141  0.25725
## log_gdppc    4.953e+00  1.004e+00   4.935 4.15e-06 ***
## population  -3.898e-09  1.307e-09  -2.981  0.00378 **
## gini        -4.364e+01  7.018e+00  -6.218 2.01e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.695 on 82 degrees of freedom
##   (988 observations deleted due to missingness)
## Multiple R-squared:  0.6994, Adjusted R-squared:  0.681
## F-statistic: 38.15 on 5 and 82 DF,  p-value: < 2.2e-16
```

```r
# 5. Low income countries
low_inc_lm <- lm(sdg_overall ~ spi_comp + di_score + log_gdppc + population + gini,
                       data = merged %>%
                         filter(income_level == "L"))
summary(low_inc_lm)
```

```
##
## Call:
## lm(formula = sdg_overall ~ spi_comp + di_score + log_gdppc +
##     population + gini, data = merged %>% filter(income_level ==
##     "L"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9199 -1.8448  0.5427  2.0511  8.9302
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.905e+01  1.444e+01   2.011   0.0567 .
## spi_comp     9.145e-02  9.677e-02   0.945   0.3549
## di_score     4.846e-01  6.452e-01   0.751   0.4605
## log_gdppc    2.996e+00  1.847e+00   1.622   0.1190
## population  -2.685e-08  4.791e-08  -0.560   0.5809
```

```
## gini        -5.343e+00  1.870e+01  -0.286   0.7778
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.184 on 22 degrees of freedom
##   (893 observations deleted due to missingness)
## Multiple R-squared:  0.2128, Adjusted R-squared:  0.03392
## F-statistic:  1.19 on 5 and 22 DF,  p-value: 0.3463
```

Extract coefficients and statistics of subgroups to comparison table

```
# Creating mod_compare_tab - comparison table
mod_compare_tab <- data.frame(
  model = c("Overall", "High Income", "Upper-Middle Income",
            "Lower-Middle Income", "Low Income"),
  coefficient = c(
    coef(overall_lm)["spi_comp"],
    coef(high_inc_lm)["spi_comp"],
    coef(upper_mid_lm)["spi_comp"],
    coef(lower_mid_lm)["spi_comp"],
    coef(low_inc_lm)["spi_comp"]
  ),

  intercept = c(
    coef(overall_lm)["(Intercept)"],
    coef(high_inc_lm)["(Intercept)"],
    coef(upper_mid_lm)["(Intercept)"],
    coef(lower_mid_lm)["(Intercept)"],
    coef(low_inc_lm)["(Intercept)"]
  ),

  Std_error = c(
  summary(overall_lm)$coefficients["spi_comp", 'Std. Error'], # just standard errors
  summary(high_inc_lm)$coefficients["spi_comp", 'Std. Error'], # just standard errors
  summary(upper_mid_lm)$coefficients["spi_comp", 'Std. Error'], # just standard errors
  summary(lower_mid_lm)$coefficients["spi_comp", 'Std. Error'], # just standard errors
  summary(low_inc_lm)$coefficients["spi_comp", 'Std. Error'] # just standard errors
  ),

  t_value = c(
    summary(overall_lm)$coefficients["spi_comp", "t value"],
    summary(high_inc_lm)$coefficients["spi_comp", "t value"],
    summary(upper_mid_lm)$coefficients["spi_comp", "t value"],
    summary(lower_mid_lm)$coefficients["spi_comp", "t value"],
    summary(low_inc_lm)$coefficients["spi_comp", "t value"]
  ),

  p_value = c(
    summary(overall_lm)$coefficients["spi_comp", "Pr(>|t|)"],
    summary(high_inc_lm)$coefficients["spi_comp", "Pr(>|t|)"],
    summary(upper_mid_lm)$coefficients["spi_comp", "Pr(>|t|)"],
    summary(lower_mid_lm)$coefficients["spi_comp", "Pr(>|t|)"],
    summary(low_inc_lm)$coefficients["spi_comp", "Pr(>|t|)"]
  ),
```

```
  r_squared = c(
    summary(overall_lm)$r.squared,
    summary(high_inc_lm)$r.squared,
    summary(upper_mid_lm)$r.squared,
    summary(lower_mid_lm)$r.squared,
    summary(low_inc_lm)$r.squared
  ),

  adj_r_squared = c(
    summary(overall_lm)$adj.r.squared,
    summary(high_inc_lm)$adj.r.squared,
    summary(upper_mid_lm)$adj.r.squared,
    summary(lower_mid_lm)$adj.r.squared,
    summary(low_inc_lm)$adj.r.squared
  ),

  n_obs = c(
    nobs(overall_lm),
    nobs(high_inc_lm),
    nobs(upper_mid_lm),
    nobs(lower_mid_lm),
    nobs(low_inc_lm)
  )
)

# Defining function for significance stars based on p-values
sig_stars <- function(p_value) {
  if (p_value <= 0.001) {
    return("***")
  } else if (p_value <= 0.01) {
    return("**")
  } else if (p_value <= 0.05) {
    return("*")
  } else if (p_value <= 0.1) {
    return(".")
  } else {
    return("")
  }
}

# make table and round for better display
mod_compare_tab <- mod_compare_tab %>%
  mutate(
    coefficient = round(coefficient, 3),
    intercept = round(intercept, 3),
    Std_error = round(Std_error, 3),
    t_value = round(t_value, 3),
    p_value = p_value,
    significance = sapply(p_value, sig_stars), #significance stars
    r_squared = round(r_squared, 3),
    adj_r_squared = round(adj_r_squared, 3)
  )
```

```r
# Print the comparison table
print(mod_compare_tab)
```

```
##                 model coefficient intercept Std_error t_value      p_value
## 1             Overall       0.247    34.627     0.020  12.309 2.552778e-30
## 2         High Income       0.189    84.314     0.025   7.424 3.149172e-12
## 3 Upper-Middle Income       0.120    60.401     0.019   6.390 2.301728e-09
## 4 Lower-Middle Income       0.267    27.306     0.051   5.201 1.430870e-06
## 5          Low Income       0.091    29.050     0.097   0.945 3.549196e-01
##   r_squared adj_r_squared n_obs significance
## 1     0.792         0.790   469          ***
## 2     0.664         0.655   207          ***
## 3     0.539         0.523   146          ***
## 4     0.699         0.681    88          ***
## 5     0.213         0.034    28
```

```r
# export mod_compare_tab
# write.csv(mod_compare_tab, file = "ols_model_comparison.csv", row.names=F)
```

##Visualizing Slopes: plotting multiple regression - by subgroup

```r
viz_gni_class <- ggplot(data = merged, aes(x = spi_comp,
                                           y = sdg_overall,
                                           color = income_level_lab)) +
  geom_point(alpha = 0.25, size = 0.75) +
  # Overall regression line (black)
  geom_smooth(aes(group = 1),
              method = "lm",
              linewidth = 0.75,
              se = FALSE,
              color = "black") +
  # Group-specific regression lines
  geom_smooth(method = "lm",
              linewidth = 0.65,
              se = FALSE) +
  scale_color_manual(
    values = c("High Income Countries" = "#1D6A96",
               "Upper-Middle Income Countries" = "#4CB5AE",
               "Lower-Middle Income Countries" = "#F3A738",
               "Low Income Countries" = "#C02942")
  ) +
  labs(title = "Relationship between SPI and SDG by World Bank Income Classification",
       x = "Statistical Performance Indicators (SPI)",
       y = "Sustainable Development Goals (SDG)",
       color = "Income Classification") +
  theme_bw()

viz_gni_class
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

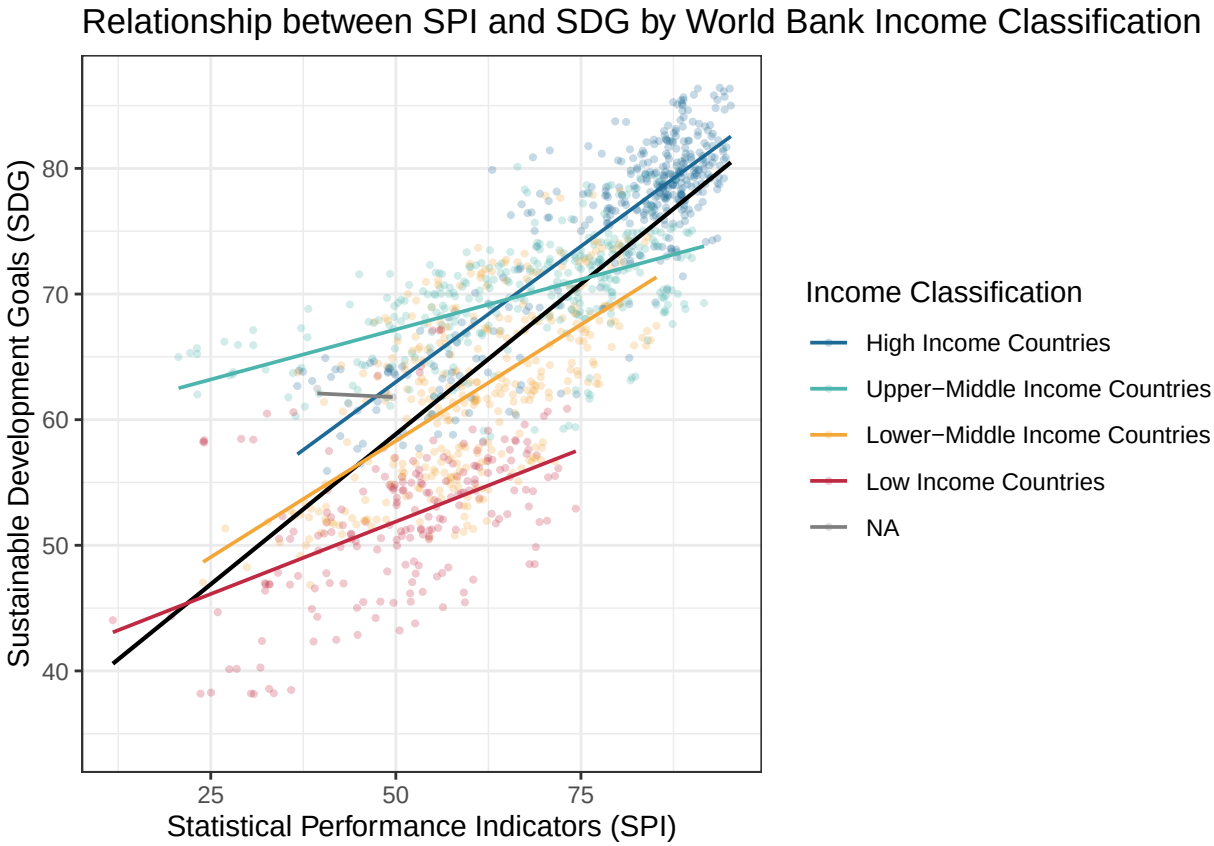
```
## Warning: Removed 3068 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 3068 rows containing non-finite outside the scale range
```

```
## (`stat_smooth()`).
```

```
## Warning: Removed 3068 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
#ggplotly(viz_gni_class)

# Save to specific folder
# ggsave("~/Documents/GitHub/QMSS_Thesis_Sanchez/Output_CSVs/fd_plot.png", p, width = 10, height = 6)
```

Mediation analysis

To test if SPI mediates the relationship between regime type and SDG outcomes: Democratic backsliding → reduces SPI → slows SDG progress

H0: SPI DOES NOT mediate (indirectly effect) the relationship between regime type and SDG status H1: SPI mediates (indirectly effects) the relationship between regime type and SDG status

- ACME (Average Causal Mediation Effect): SPI's indirect effect.
- ADE (Average Direct Effect): Regime type's direct effect, excluding SPI

```
library(mediation)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
```

```
##     expand, pack, unpack
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: sandwich
```

```
## mediation: Causal Mediation Analysis
## Version: 4.5.0
```

```
#Total Effect: Check if regime type directly affects SDG scores (without SPI)
lm_sdg <- lm(sdg_overall ~ di_score + log_gdppc + population, data = merged)
summary(lm_sdg)
```

```
##
## Call:
## lm(formula = sdg_overall ~ di_score + log_gdppc + population,
##     data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.6744  -4.2531  -0.1881   4.2867  17.4892
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.028e+01  7.550e-01  26.867   <2e-16 ***
## di_score    1.385e+00  7.387e-02  18.751   <2e-16 ***
## log_gdppc   4.348e+00  1.091e-01  39.836   <2e-16 ***
## population  2.433e-10  8.032e-10   0.303    0.762
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.03 on 2370 degrees of freedom
##   (1994 observations deleted due to missingness)
## Multiple R-squared:  0.6766, Adjusted R-squared:  0.6762
## F-statistic:  1653 on 3 and 2370 DF,  p-value: < 2.2e-16
```

```
#Mediator model: Check if regime type affects SPI
lm_spi <- lm(spi_comp ~ di_score + log_gdppc + population, data = merged)
summary(lm_spi)
```

```
##
## Call:
## lm(formula = spi_comp ~ di_score + log_gdppc + population, data = merged)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.101  -6.478   0.789   7.460  32.324
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.523e+01  2.185e+00   6.970 5.49e-12 ***
## di_score    3.463e+00  2.096e-01  16.518  < 2e-16 ***
## log_gdppc   3.706e+00  3.156e-01  11.741  < 2e-16 ***
## population  4.937e-09  2.132e-09   2.315   0.0208 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 11.3 on 1079 degrees of freedom
##   (3285 observations deleted due to missingness)
## Multiple R-squared:  0.5248, Adjusted R-squared:  0.5235
## F-statistic: 397.2 on 3 and 1079 DF,  p-value: < 2.2e-16
```

```
#outcome model: Check if SPI affects SDG scores while controlling for regime type
lm_sdg_controlled <- lm(sdg_overall ~ spi_comp + di_score + log_gdppc + population, data = merged)
summary(lm_sdg_controlled)
```

```
##
## Call:
## lm(formula = sdg_overall ~ spi_comp + di_score + log_gdppc +
##     population, data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.1291  -3.3677   0.0302   3.1003  12.4129
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.873e+01  9.579e-01  19.549  < 2e-16 ***
## spi_comp     2.642e-01  1.306e-02  20.232  < 2e-16 ***
## di_score     3.307e-01  1.006e-01   3.286  0.00105 **
## log_gdppc    3.330e+00  1.438e-01  23.164  < 2e-16 ***
## population  -1.123e-09  9.169e-10  -1.224  0.22109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.847 on 1078 degrees of freedom
##   (3285 observations deleted due to missingness)
## Multiple R-squared:  0.7697, Adjusted R-squared:  0.7688
## F-statistic: 900.7 on 4 and 1078 DF,  p-value: < 2.2e-16
```

```
#Mediation test: Quantify how much of regime type's effect on SDGs operates through SPI
med_model <- mediate(lm_spi, lm_sdg_controlled, treat = "di_score", mediator = "spi_comp")
summary(med_model)
```

```
##
## Causal Mediation Analysis
##
## Quasi-Bayesian Confidence Intervals
##
##                Estimate 95% CI Lower 95% CI Upper p-value
## ACME              0.914        0.780         1.05  <2e-16 ***
## ADE               0.333        0.141         0.54   0.002 **
## Total Effect      1.247        1.035         1.46  <2e-16 ***
## Prop. Mediated    0.733        0.615         0.87  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 1083
##
##
## Simulations: 1000
```

ACEM: SPI's indirect effect = 0.917 units ADE: Regime type's direct effect, excluding SPI = 0.326 units

Total Effect: = 1.243 units Proportion Mediated: = 74.1% of total units

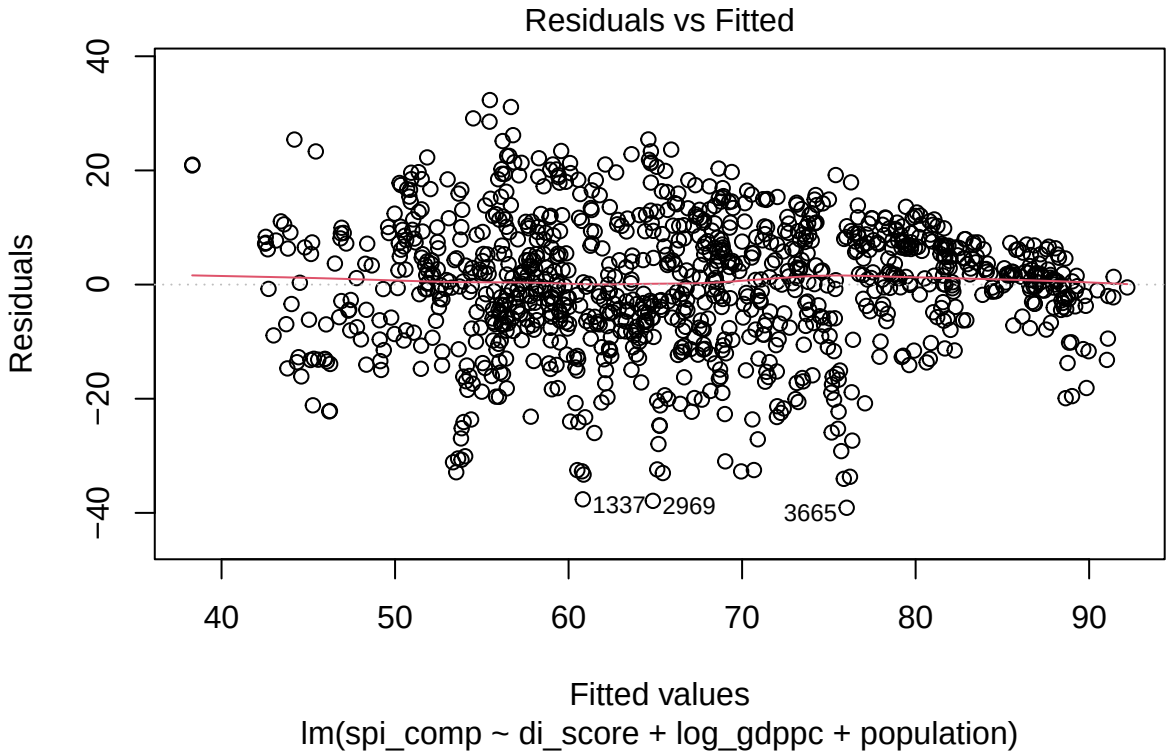
Interpretation: A 1-unit DI increase boosts SDG scores by 1.243 total units, with 0.917 units (~74% of units) transmitted through SPI. The remaining 0.326 units reflect direct DI effects (e.g., governance reforms unrelated to statistics).

Because the ACME (indirect effect of SPI on sdg_overall) is highly significant ($p < 0.001$), SPI mediates the regime-SDG relationship, based on the model.

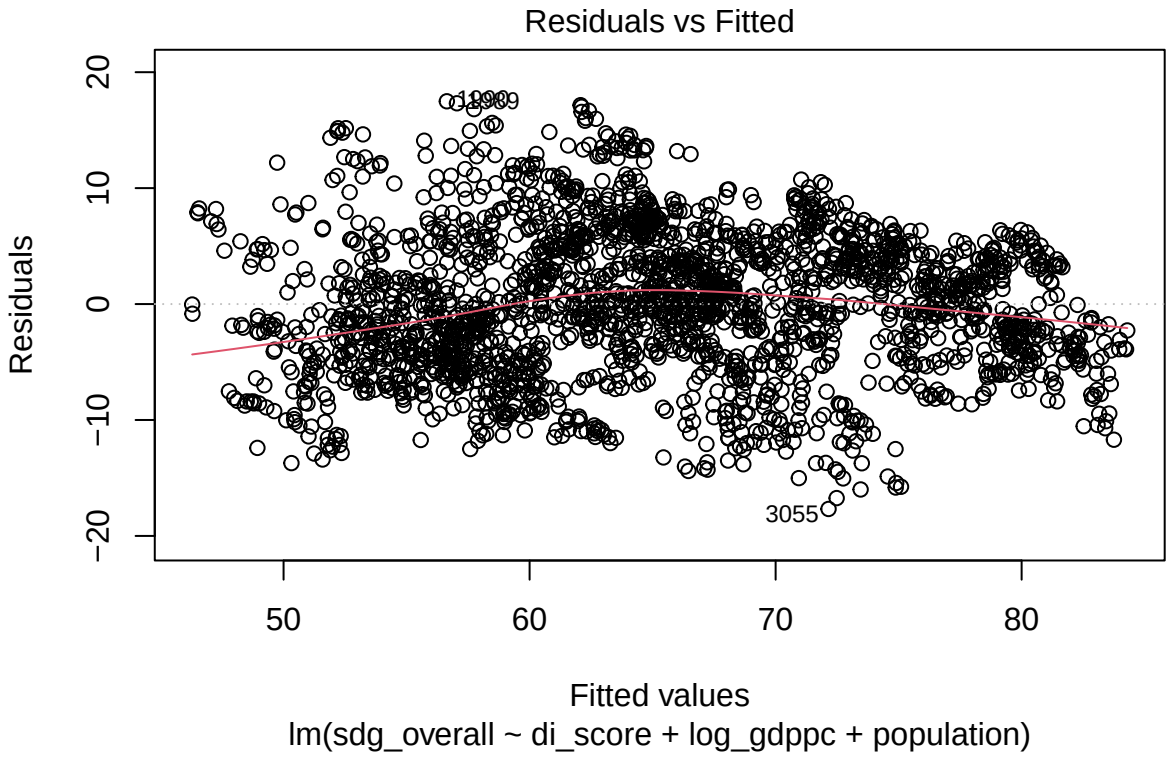
Because the ADE (the direct effect between di_score on sdg_overall) is also significant ($p = 0.002$), although much less than the ACME estimate, SPI DOES NOT FULLY explain the connection between regime type and sdg status, based on the model.

Linearity vs non-linearity

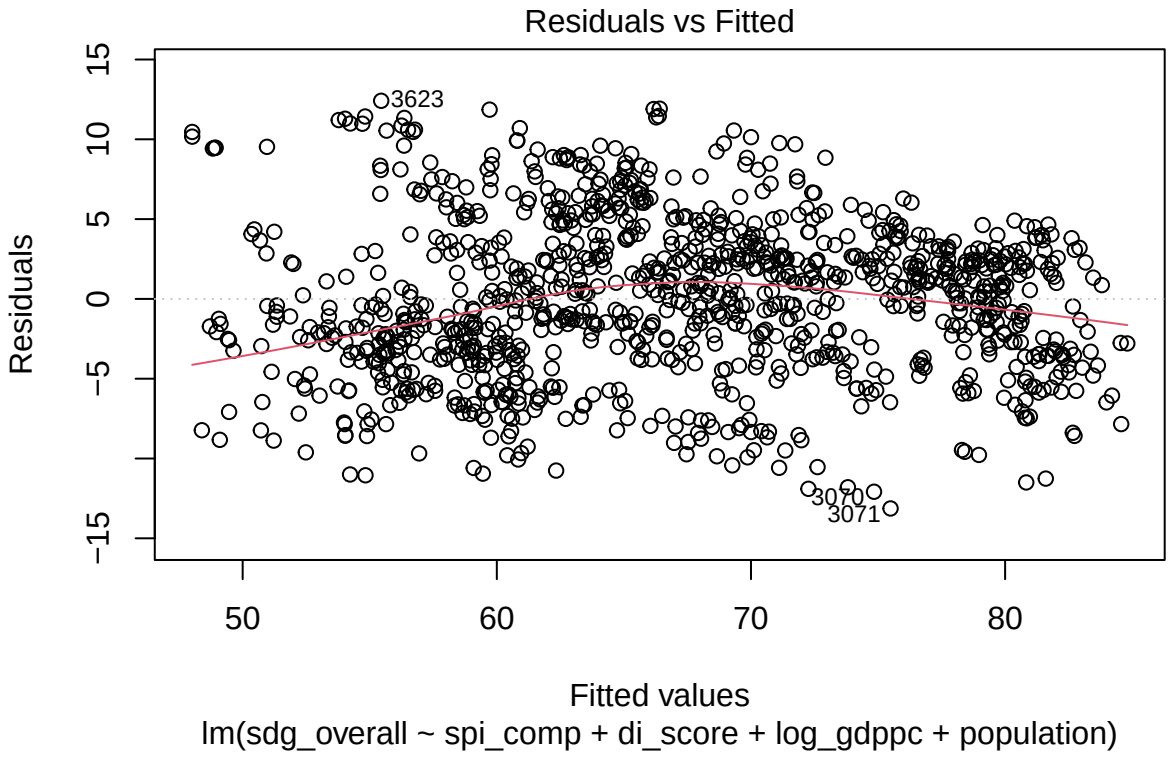
```
plot(lm_spi, which = 1)  # residuals for SPI model
```




```
plot(lm_sdg, which = 1)  # residuals for SDG model
```




```
plot(lm_sdg_controlled, which = 1)  # residuals for SDG model controlled
```




```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
```

```
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
resettest(lm_spi, power = 2:3, type = "fitted")
```

```
##
##  RESET test
##
## data:  lm_spi
## RESET = 1.9819, df1 = 2, df2 = 1077, p-value = 0.1383
```

```
resettest(lm_sdg, power = 2:3, type = "fitted")
```

```
##
##  RESET test
##
## data:  lm_sdg
## RESET = 40.814, df1 = 2, df2 = 2368, p-value < 2.2e-16
```

```
resettest(lm_sdg_controlled, power = 2:3, type = "fitted")
```

```
##
##  RESET test
##
## data:  lm_sdg_controlled
## RESET = 21.023, df1 = 2, df2 = 1076, p-value = 1.106e-09
```

```
#validate with sensitivity test
#sensmediation::sensmed(model_m, model_y, sims = 500)
```

STILL LEFT TO INCLUDE

- Use Robust Standard Errors across all models
- Include GNI Coefficient control in all models
- Update results based on these changes