Component 1: Multiple OLS, Interactions, Subgroups

Sevastian Sanchez

2025-05-05

All Countries, Preliminary Analysis (SPI x SDGs)

#FIRST: Libraries, Directory & Data

#SECOND: Run function in r-script: df_years_function.R
[ADJUST TIME OR SKIP AND LOADING DATA FROM DIRECTORY]

THIRD: Load and Refine Data

[SKIP IF LOADING FROM DIRECTORY]

FOURTH: Load cleaned 'merged' Dataset

[ADJUST VARIABLES OR SKIP IF LOADING FROM CSV]

FINALLY: LOAD FINAL MERGED CSV

```r
#load final merged df
merged <- read_csv("data/Main CSV Outputs/merged_final_df.csv")
```

```
## Rows: 3360 Columns: 40
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): country_name, country_code, income_level, income_level_lab
## dbl (36): year, year_fct, sdg_overall, spi_comp, sci_overall, di_score, regi...
##
## 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.
```

COMPONENT 1: 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

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

#x-var 1 = spi
correlation_sdg_spi <- cor(merged$sdg_overall, merged$spi_comp, use = "complete.obs")^2

#x-var 2 = sci
```

```r
correlation_sdg_sci <- cor(merged$sdg_overall, merged$sci_overall, use = "complete.obs")^2

#x-var 3 = di
correlation_sdg_di <- cor(merged$sdg_overall, merged$di_score, use = "complete.obs")^2

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

```
## [1] "Correlation coefficient: 0.616037202309322 (SPI) 0.410940651230861 (SCI) 0.452968121616442 (DI)
```

Correlation coefficient/R-sq (SPI): 0.616037202309322
Correlation coefficient/R-sq (SCI): 0.410940651230861 Correlation coefficient/R-sq (DI): 0.452968121616442

NAIVE OLS: Comparing SPI & SCI w/o controls

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

```r
# 2. OLS for SPI and SDG - Overall
ols_spi_naive <- lm(sdg_overall ~ spi_comp, data = merged)
summary(ols_spi_naive)
```

```
##
## Call:
## lm(formula = sdg_overall ~ spi_comp, data = merged)
##
## 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 ***
## spi_comp     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
##   (2060 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
```

```r
# 2. OLS for SCI and SDG - Overall
ols_sci_naive <- lm(sdg_overall ~ sci_overall, data = merged)
summary(ols_sci_naive)
```

```
##
## Call:
## lm(formula = sdg_overall ~ sci_overall, data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.9250  -4.9781   0.2876   4.8825  18.8727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 33.87135    0.71668   47.26   <2e-16 ***
## sci_overall  0.39081    0.01028   38.00   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.218 on 2070 degrees of freedom
##   (1288 observations deleted due to missingness)
## Multiple R-squared:  0.4109, Adjusted R-squared:  0.4107
## F-statistic:  1444 on 1 and 2070 DF,  p-value: < 2.2e-16
```

```r
# 3. Multiple Regression with both SPI and SCI
ols_multiple_naive <- lm(sdg_overall ~ spi_comp + sci_overall, data = merged)
summary(ols_multiple_naive)
```

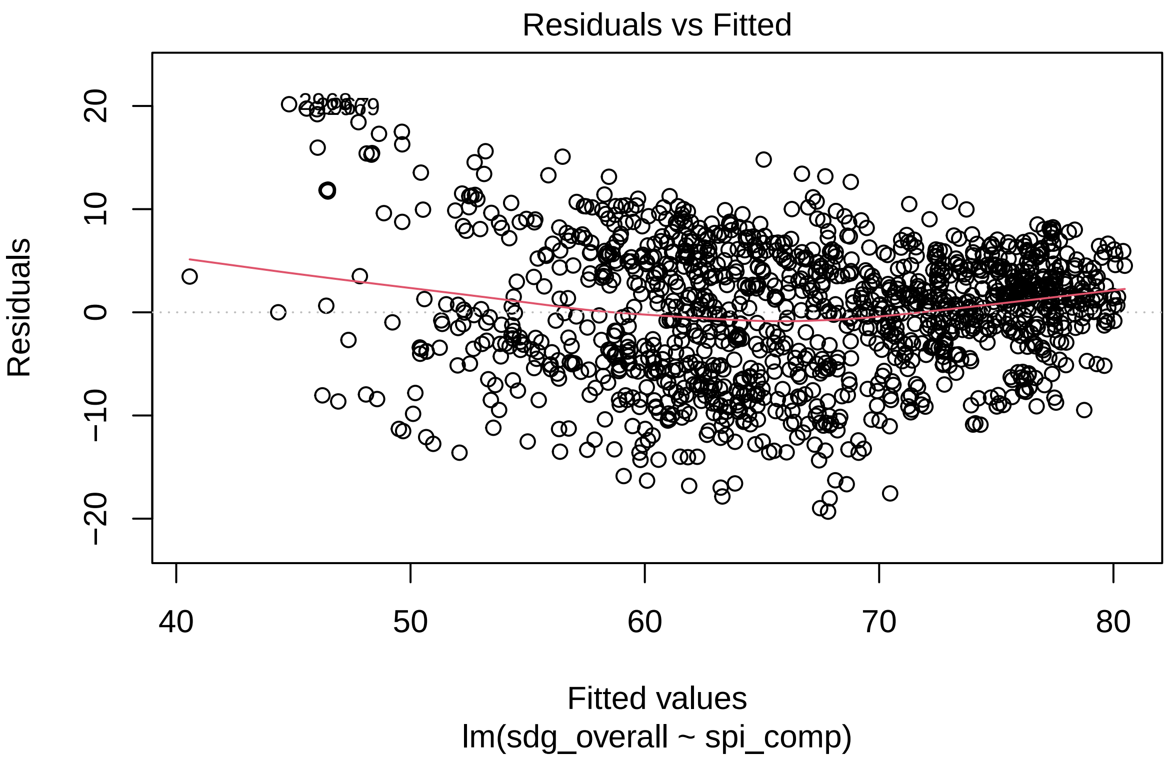
```
##
## Call:
## lm(formula = sdg_overall ~ spi_comp + sci_overall, data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.5483  -5.4484   0.4037   4.7941  17.9050
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.86438    1.27744  28.075  < 2e-16 ***
## spi_comp     0.28779    0.03369   8.542  < 2e-16 ***
## sci_overall  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
##   (2764 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: 0.39081 (p-value < 0.001)
ols_multiple_naive: spi: 0.28779 (p-value < 0.001); sci: 0.15311 (p-value < 0.001)

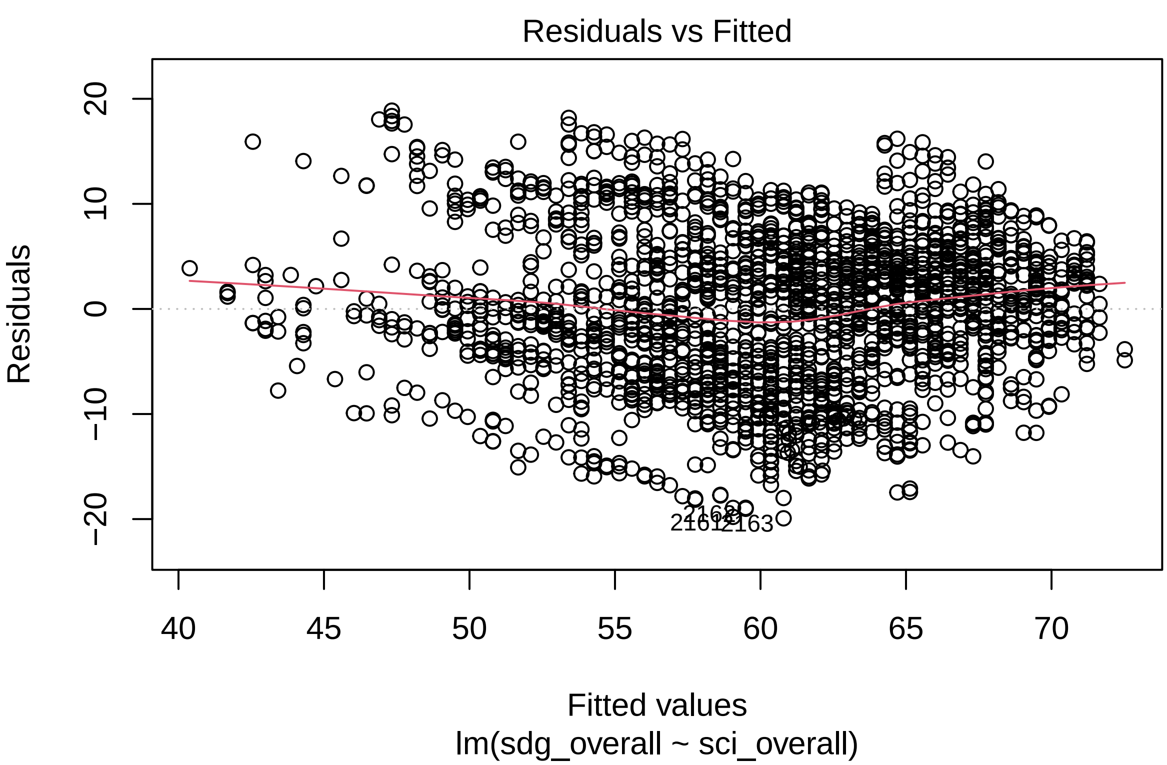
The impact of SCI on SDG and SPI on SDG are statistically significant, in all models. SPI appears to have a greater impact on SDGs compared to that of SCI, regardless of the model. All of this is without controls.

#Checking for Heteroskedasticity: residual plots

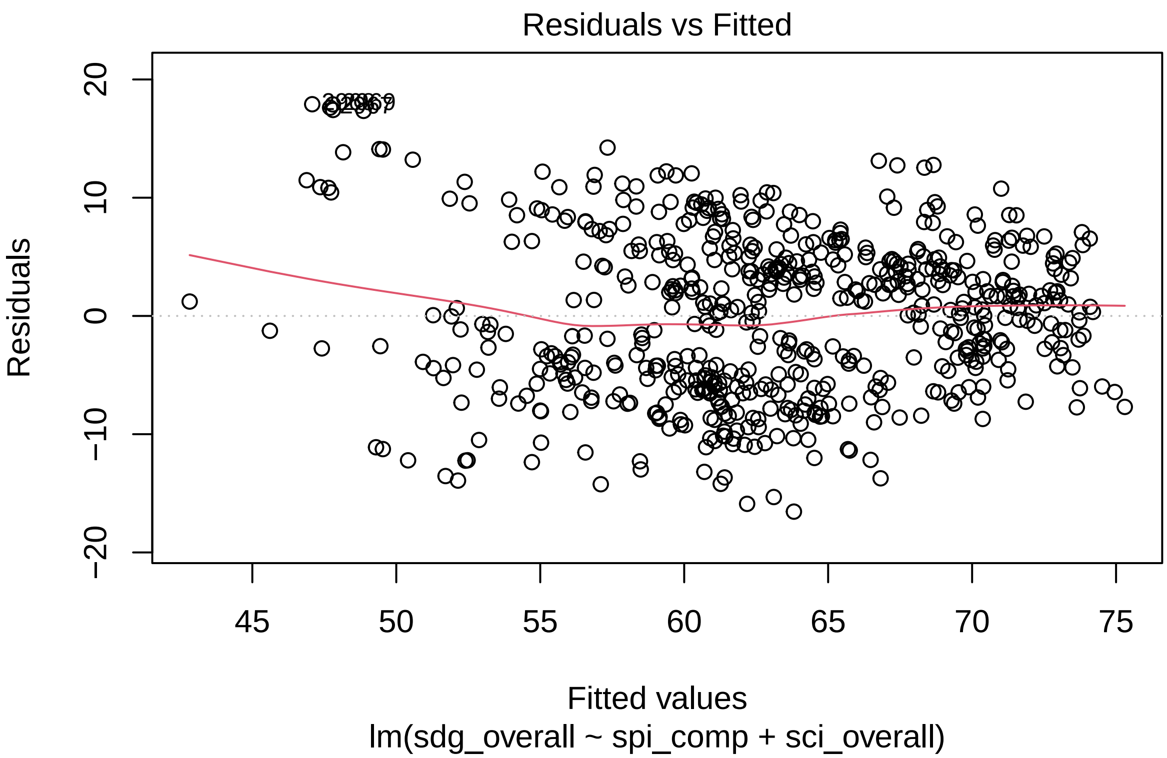
```r
#residual plots
plot(ols_spi_naive, which = 1)  # SPI model
```




```r
plot(ols_sci_naive, which = 1)   # SDG model
```




```r
plot(ols_multiple_naive, which = 1)   # SDG model controlled
```



U-shaped residuals detected, suggests non-linearity of x-variable terms. Additional tests reconfirm non-linearity (See Breusch-Pagan Test below).

TEST 1: Comparing SPI & SCI WITH controls AND Robust Standard Errors

Applying controls and robust (Huber-White) standard errors

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

```
# 1. OLS for SPI and SDG - Overall
ols_spi <- lm(sdg_overall ~ spi_comp + log_gdppc + population +  di_score + year_fct, data = merged)
summary(ols_spi)
```

```
##
## Call:
## lm(formula = sdg_overall ~ spi_comp + log_gdppc + population +
##     di_score + year_fct, data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.2515  -3.3258   0.0273   3.2133  13.3604
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.024e+02  1.597e+02   4.398 1.20e-05 ***
## spi_comp     2.864e-01  1.396e-02  20.522  < 2e-16 ***
## log_gdppc    3.300e+00  1.428e-01  23.112  < 2e-16 ***
## population  -1.215e-09  9.099e-10  -1.336   0.1820
## di_score     2.172e-01  1.033e-01   2.103   0.0357 *
## year_fct    -3.389e-01  7.916e-02  -4.281 2.02e-05 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.808 on 1077 degrees of freedom
##   (2277 observations deleted due to missingness)
## Multiple R-squared:  0.7736, Adjusted R-squared:  0.7725
## F-statistic: 735.8 on 5 and 1077 DF,  p-value: < 2.2e-16
```

```r
coeftest(ols_spi, vcov = vcovHC(ols_spi, type = "HC1"))
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  7.0238e+02  1.5950e+02  4.4036 1.171e-05 ***
## spi_comp     2.8641e-01  1.5627e-02 18.3280 < 2.2e-16 ***
## log_gdppc    3.3001e+00  1.8662e-01 17.6836 < 2.2e-16 ***
## population  -1.2152e-09  8.4979e-10 -1.4300   0.15302
## di_score     2.1725e-01  1.1547e-01  1.8814   0.06019 .
## year_fct    -3.3890e-01  7.9041e-02 -4.2877 1.967e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
##
## Call:
## lm(formula = sdg_overall ~ sci_overall + log_gdppc + population +
##     di_score + year_fct, data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.4270  -3.0281  -0.1157   3.0529  14.5665
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.365e+02  5.997e+01  -8.946  < 2e-16 ***
## sci_overall  2.398e-01  9.689e-03  24.747  < 2e-16 ***
## log_gdppc    5.409e+00  1.414e-01  38.261  < 2e-16 ***
## population  -2.189e-09  6.988e-10  -3.132  0.00177 **
## di_score    -6.646e-02  8.427e-02  -0.789  0.43045
## year_fct     2.678e-01  2.983e-02   8.977  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.725 on 1508 degrees of freedom
##   (1846 observations deleted due to missingness)
## Multiple R-squared:  0.7391, Adjusted R-squared:  0.7383
## F-statistic: 854.5 on 5 and 1508 DF,  p-value: < 2.2e-16
```

```r
coeftest(ols_sci, vcov = vcovHC(ols_sci, type = "HC1"))
```

```
##
## t test of coefficients:
##
```

```
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept) -5.3654e+02  6.2002e+01 -8.6536 < 2.2e-16 ***
## sci_overall  2.3976e-01  1.0273e-02 23.3390 < 2.2e-16 ***
## log_gdppc    5.4092e+00  1.4521e-01 37.2496 < 2.2e-16 ***
## population  -2.1890e-09  4.1473e-10 -5.2782 1.495e-07 ***
## di_score    -6.6457e-02  8.6120e-02 -0.7717    0.4404
## year_fct     2.6778e-01  3.0816e-02  8.6895 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# 3. Multiple Regression with both SPI and SCI
ols_multiple <- lm(sdg_overall ~ spi_comp + sci_overall + log_gdppc + population + di_score + year_fct,
summary(ols_multiple)
```

```
##
## Call:
## lm(formula = sdg_overall ~ spi_comp + sci_overall + log_gdppc +
##     population + di_score + year_fct, data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2344  -2.6877  -0.0114   2.5179  12.9086
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.465e+02  2.953e+02  -1.851  0.06474 .
## spi_comp     1.195e-01  2.671e-02   4.472 9.37e-06 ***
## sci_overall  1.488e-01  2.398e-02   6.202 1.08e-09 ***
## log_gdppc    5.809e+00  2.207e-01  26.323  < 2e-16 ***
## population  -3.140e-09  1.006e-09  -3.122  0.00189 **
## di_score    -4.158e-01  1.312e-01  -3.169  0.00161 **
## year_fct     2.718e-01  1.463e-01   1.858  0.06371 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.328 on 560 degrees of freedom
##   (2793 observations deleted due to missingness)
## Multiple R-squared:  0.7639, Adjusted R-squared:  0.7614
## F-statistic:   302 on 6 and 560 DF,  p-value: < 2.2e-16
```

```
coeftest(ols_multiple, vcov = vcovHC(ols_multiple, type = "HC1"))
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept) -5.4648e+02  3.0198e+02 -1.8096  0.070889 .
## spi_comp     1.1945e-01  2.9238e-02  4.0855 5.041e-05 ***
## sci_overall  1.4876e-01  2.6159e-02  5.6869 2.085e-08 ***
## log_gdppc    5.8088e+00  2.1022e-01 27.6322 < 2.2e-16 ***
## population  -3.1398e-09  5.0026e-10 -6.2763 6.953e-10 ***
## di_score    -4.1576e-01  1.2736e-01 -3.2644  0.001164 **
## year_fct     2.7179e-01  1.4966e-01  1.8161  0.069890 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Creating dataframe combining summary and Robust SE results

```r
# For model statistics extraction
summary_spi <- summary(ols_spi)
summary_sci <- summary(ols_sci)
summary_multiple <- summary(ols_multiple)

# extracting robust SEs and coefficients (using coeftest)
rob_stats_spi <- coeftest(ols_spi, vcov = vcovHC(ols_spi, type = "HC1"))
rob_stats_sci <- coeftest(ols_sci, vcov = vcovHC(ols_sci, type = "HC1"))
rob_stats_multiple <- coeftest(ols_multiple, vcov = vcovHC(ols_multiple, type = "HC1"))

#SPI Statistics DF
spi_df <- data.frame(
  model = "M1: ols_spi",
  term = rownames(rob_stats_spi),
  estimate = rob_stats_spi[, 1],
  std.error = rob_stats_spi[, 2],
  t.statistic = rob_stats_spi[, 3],
  p.value = rob_stats_spi[, 4],
  residual.SE = summary_spi$sigma,
  r.squared = summary_spi$r.squared,
  adj.r.squared = summary_spi$adj.r.squared,
  row.names = NULL
)

#SCI Statistics DF
sci_df <- data.frame(
  model = "M2: ols_sci",
  term = rownames(rob_stats_sci),
  estimate = rob_stats_sci[, 1],
  std.error = rob_stats_sci[, 2],
  t.statistic = rob_stats_sci[, 3],
  p.value = rob_stats_sci[, 4],
  residual.SE = summary_sci$sigma,
  r.squared = summary_sci$r.squared,
  adj.r.squared = summary_sci$adj.r.squared,
  row.names = NULL
)

#Combined Mod Statistics DF
multiple_df <- data.frame(
  model = "M3: ols_multiple",
  term = rownames(rob_stats_multiple),
  estimate = rob_stats_multiple[, 1],
  std.error = rob_stats_multiple[, 2],
  t.statistic = rob_stats_multiple[, 3],
  p.value = rob_stats_multiple[, 4],
  residual.SE = summary_multiple$sigma,
  r.squared = summary_multiple$r.squared,
  adj.r.squared = summary_multiple$adj.r.squared,
  row.names = NULL
)
```

```r
# Bind all together into one tidy dataframe
robust_mods_df <- bind_rows(spi_df, sci_df, multiple_df)

# Attributes under column names
attr(robust_mods_df$std.error, "label") <- "Robust Std. Errors Adjusted"
attr(robust_mods_df$t.statistic, "label") <- "Robust Std. Errors Adjusted"
attr(robust_mods_df$p.value, "label") <- "Robust Std. Errors Adjusted"

#save to output_CSVs folder
write.csv(robust_mods_df, file = "output_CSVs/ols_mods_results.csv")

# View the result
print(robust_mods_df)
```

```
##               model        term      estimate    std.error t.statistic
## 1       M1: ols_spi (Intercept)  7.023774e+02 1.595014e+02   4.4035822
## 2       M1: ols_spi    spi_comp  2.864147e-01 1.562713e-02  18.3280441
## 3       M1: ols_spi   log_gdppc  3.300063e+00 1.866166e-01  17.6836478
## 4       M1: ols_spi  population -1.215172e-09 8.497912e-10  -1.4299653
## 5       M1: ols_spi    di_score  2.172487e-01 1.154748e-01   1.8813514
## 6       M1: ols_spi    year_fct -3.389012e-01 7.904091e-02  -4.2876677
## 7       M2: ols_sci (Intercept) -5.365395e+02 6.200174e+01  -8.6536213
## 8       M2: ols_sci sci_overall  2.397649e-01 1.027312e-02  23.3390490
## 9       M2: ols_sci   log_gdppc  5.409203e+00 1.452150e-01  37.2496201
## 10      M2: ols_sci  population -2.188985e-09 4.147255e-10  -5.2781523
## 11      M2: ols_sci    di_score -6.645668e-02 8.611952e-02  -0.7716796
## 12      M2: ols_sci    year_fct  2.677751e-01 3.081609e-02   8.6894580
## 13 M3: ols_multiple (Intercept) -5.464758e+02 3.019820e+02  -1.8096303
## 14 M3: ols_multiple    spi_comp  1.194510e-01 2.923794e-02   4.0854790
## 15 M3: ols_multiple sci_overall  1.487645e-01 2.615911e-02   5.6869121
## 16 M3: ols_multiple   log_gdppc  5.808824e+00 2.102190e-01  27.6322487
## 17 M3: ols_multiple  population -3.139784e-09 5.002640e-10  -6.2762541
## 18 M3: ols_multiple    di_score -4.157593e-01 1.273603e-01  -3.2644340
## 19 M3: ols_multiple    year_fct  2.717938e-01 1.496580e-01   1.8160997
##          p.value residual.SE r.squared adj.r.squared
## 1   1.170908e-05    4.808283 0.7735523     0.7725010
## 2   1.609530e-65    4.808283 0.7735523     0.7725010
## 3   1.234510e-61    4.808283 0.7735523     0.7725010
## 4   1.530170e-01    4.808283 0.7735523     0.7725010
## 5   6.019368e-02    4.808283 0.7735523     0.7725010
## 6   1.967494e-05    4.808283 0.7735523     0.7725010
## 7   1.257018e-17    4.724652 0.7391159     0.7382509
## 8  4.162604e-103    4.724652 0.7391159     0.7382509
## 9  6.970604e-216    4.724652 0.7391159     0.7382509
## 10  1.495226e-07    4.724652 0.7391159     0.7382509
## 11  4.404251e-01    4.724652 0.7391159     0.7382509
## 12  9.313776e-18    4.724652 0.7391159     0.7382509
## 13  7.088915e-02    4.327524 0.7639006     0.7613710
## 14  5.041114e-05    4.327524 0.7639006     0.7613710
## 15  2.085117e-08    4.327524 0.7639006     0.7613710
## 16 1.129016e-106    4.327524 0.7639006     0.7613710
## 17  6.952679e-10    4.327524 0.7639006     0.7613710
## 18  1.163770e-03    4.327524 0.7639006     0.7613710
## 19  6.988965e-02    4.327524 0.7639006     0.7613710
```

We reject the null hypothesis that there is no relationship between SPI and SDG composite scores. Additionally, we reject the null hypothesis that there is no relationship between SCI and SDG composite scores. Holding all else constant (log GDP per capita, democracy score and population), SPI and SCI exhibit positive moderate and statistically significant relationships with SDG status.

ols_spi: 0.28641 (p-value < 0.001)
ols_sci: 0.23976 (p-value < 0.001)
ols_multiple: spi: 0.11945 (p-value < 0.001); sci: 0.14876 (p-value < 0.001)

When compared in separate models, SPI has a greater impact on SDG status (0.28641) than SCI (0.23976). This suggests that a one-unit increase in SPI is associated with a larger improvement in SDG outcomes compared to a one-unit increase in SCI, holding all controls constant.

Interestingly, the opposite holds true in a multiple regression model containing both SPI and SCI. SPI's impact on SDG status (0.11945) (net of SPI) is less than that of SCI's (0.14876) (net of SCI), holding all controls constant. When together, the coefficients represent the unique impact of each predictor variable (measures of statistical capacity) 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 both coefficients as seen in model 3: 0.28641 to 0.11945 for SPI (58.29% decrease); and from 0.23976 to 0.14876 for SCI (37.95% decrease). Such indicates that they're both capturing much of the same underlying relationship with SDG status.

However, the fact that both SPI and SCI remain significant when included together (model 3) with a high adjusted R-sq (0.7614) suggests that they capture different dimensions of statistical capacity that independently contribute to SDG status.

Checking for Multicolinearity: VIF of SPI & SCI

```r
# Check correlation between SPI and SCI
cor(merged$spi_comp, merged$sci_overall, use = "complete.obs")
```

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

```r
# Check VIF (Variance Inflation Factor) in Model 3
vif(ols_multiple)
```

```
##    spi_comp sci_overall   log_gdppc  population    di_score    year_fct
##    4.287175    3.790699    1.484531    1.029921    1.603428    1.288969
```

```r
#make into Datatable
vif_vals <- vif(ols_multiple)   # returns a named vector
tidy_vif <- enframe(vif_vals, name = "term", value = "vif")
print(tidy_vif)
```

```
## # A tibble: 6 x 2
##   term          vif
##   <chr>       <dbl>
## 1 spi_comp     4.29
## 2 sci_overall  3.79
## 3 log_gdppc    1.48
## 4 population   1.03
## 5 di_score     1.60
## 6 year_fct     1.29
```

colinearity: The correlation between SCI and SPI is about 0.8277. When placed within the same model, SCI inflated the standard error of SPI from 0.01396 to 0.02671. SCI had a similar reaction from the SPI with its standard error increasing from 0.00969 to 0.02398.

VIF: Such multicollinearity 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).

VIF Results: term vif 1 spi_comp 4.29 2 sci_overall 3.79 3 log_gdppc 1.48 4 population 1.03 5 di_score 1.60 6 year_fct 1.29 (categorical)

Overall there reveals no severe multicollinearity (all GVIF < 5). There is moderate correlation between statistical capacity measures (spi_comp and sci_overall) with SPI moderately inflated by a factor of 4.29 and SCI inflated by a factor of 3.79. Nevertheless, it is acceptable to include both in the same model as doing so will not severely impact estimates with both factors less than 5.0. Even so, there are significant limitations in either model that warrant strong consideration, including sample size, and longitudinal suitability. All other variables show minimal multicollinearity concerns.

Checking misspecification missing non-linear or omitted interactions

```r
#lm test package
resettest(ols_spi, power = 2:3, type = "fitted")
```

```
##
##  RESET test
##
## data:  ols_spi
## RESET = 19.743, df1 = 2, df2 = 1075, p-value = 3.795e-09
```

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

```
##
##  RESET test
##
## data:  ols_sci
## RESET = 43.023, df1 = 2, df2 = 1506, p-value < 2.2e-16
```

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

```
##
##  RESET test
##
## data:  ols_multiple
## RESET = 16.532, df1 = 2, df2 = 558, p-value = 1.059e-07
```

All three models show statistically significant evidence of misspecification. Given the high variablility of statistical capacity measures and control variables like GDP Per Capita and Total Population, misspecification here is likely to be the result of ommitted interaction terms or heteroskedasticity.

As pointed out by (AUTHOR) who maintained that ______. Thus, to test for the existence of non-linear omitted variables, or lack thereof, I deploy a Breusch-Pagan test for heteroskedasticity.

Checking for Heteroskedasticity: Breusch-Pagan Test

This validates the need for integrating robust standard errors in our models

```r
#Breusch-Pagan tests
bptest(ols_spi)
```

```
##
##  studentized Breusch-Pagan test
##
```

```
## data:  ols_spi
## BP = 192.11, df = 5, p-value < 2.2e-16
```

```r
bptest(ols_sci)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  ols_sci
## BP = 101.71, df = 5, p-value < 2.2e-16
```

```r
bptest(ols_multiple)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  ols_multiple
## BP = 36.535, df = 6, p-value = 2.169e-06
```

```r
#make into objects
bp_spi <- bptest(ols_spi)
bp_sci <- bptest(ols_sci)
bp_multiple <- bptest(ols_multiple)

# combine for data frame
bp_tests <- list(
  ols_spi = bp_spi,
  ols_sci = bp_sci,
  ols_multiple = bp_multiple
)

# Tidy all tests and add a "model" column
df_bptests <- bp_tests %>%
  map_df(~ tidy(.x), .id = "model")


#write.table(df_bptests, file = 'output_CSVs/df_bptests_heterosked.csv', row.names=F, sep = ",")
```

Model: BP statistic p-value ols_spi 192.11 < 2.2e-16
ols_sci 101.71 < 2.2e-16 ols_multiple 36.54 < 2.169e-06

The Breusch-Pagan Test was applied to test to see whether residuals are constant across observations, which signals unaccounted non-linear relationships, especially with macro factors such as GDP Per Capita and Population in the models. This is important because Ordinary Least Squares models assume constant error variance. In such a complex world of diverse cultural and everchanging political structures across 200 countries, cross-national data, especially in development, is rarely ever linear. Accordingly, this test evaluates the extent of such non-linearity among specified predictors.

As such, results indicate strong evidence of heteroskedasticity in all three models. The small p-values in all models indicates that the variance of residuals are not constant across observations in all three models. This reinforces the motivation behind applying robust standard errors, which have been integrated to all OLS models. Without Robust SEs, there is a risk of inflated t-statistics, leading to false significance and misinterpretation of results.

Despite the improvement from 192.11 (SPI) and 101.71 (SCI) to 36.54 (Both), there still remains statically significant heteroskedasticity in the combined model. Both statistical capacity measures create a better-specified model (ols_multiple), though not enough to eliminate heteroskedasticity entirely.

Missing Data Structure & Interpretations

Systematic, non-random missing data pattern: SPI has near complete country data coverage (165 out of 168 countries with an SDG score), but with a stubborn temporal limitation (2016-2023). On the other hand, SCI has longer temporal coverage (2004-2020), but lacks reporting on high-income countries focusing primarily on the developing world (123 out of 168 countries with an SDG score).

In model 1 (SDG ~ SPI), democracy score (di_score) is not statistically significant (-0.0243, p=0.8673). However, in model 2 (SDG ~ SCI) democracy score is highly significant (0.5556, $p < 0.001$). In model 3, with both SCI and SPI, democracy score is marginally significant (0.0301, p=0.0484). This suggests that SCI's relationship with SDG outcomes may be closely linked to regime/democratic governance. Considering the non-random missing data structured previously mentioned, the difference in significance between the models for democracy score signals an even greater need to perform subgroup analysis of countries at different stages/levels of development.

AIC/BIC Checking Fit

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

```
## Warning in AIC.default(ols_spi, ols_sci, ols_multiple): models are not all
## fitted to the same number of observations

##              df      AIC
## ols_spi       7 6482.761
## ols_sci       7 9006.394
## ols_multiple  8 3279.338
```

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

```
## Warning in BIC.default(ols_spi, ols_sci, ols_multiple): models are not all
## fitted to the same number of observations

##              df      BIC
## ols_spi       7 6517.673
## ols_sci       7 9043.652
## ols_multiple  8 3314.060
```

```r
# INTO DATAFRAME
aic_vals <- c(
  AIC(ols_spi),
  AIC(ols_sci),
  AIC(ols_multiple)
)

bic_vals <- c(
  BIC(ols_spi),
  BIC(ols_sci),
  BIC(ols_multiple)
)

# Model names
model_names <- c("ols_spi", "ols_sci", "ols_multiple")

# Combine into a dataframe
aic_bic_ols_results <- data.frame(
```

```r
  model = model_names,
  AIC = aic_vals,
  BIC = bic_vals
)

print(aic_bic_ols_results)
```

```
##          model      AIC      BIC
## 1      ols_spi 6482.761 6517.673
## 2      ols_sci 9006.394 9043.652
## 3 ols_multiple 3279.338 3314.060
```

```r
# saving to output_CSVs
write.csv(aic_bic_ols_results, file = "output_CSVs/aic_bic_ols_results.csv")
```

AIC/BIC Results AIC(ols_spi, ols_sci, ols_multiple) df AIC ols_spi 7 6482.761 ols_sci 7 9006.394 ols_multiple 8 3279.338

BIC(ols_spi, ols_sci, ols_multiple) df BIC ols_spi 7 6517.673 ols_sci 7 9043.652 ols_multiple 8 3314.060

Adjusted R-squares ols_spi: 0.7725 ols_sci: 0.7383 ols_multiple: 0.7614

Selecting Best Model

Best fit: ols_spi (Adj Rsq: 0.7725) (AIC/BIC: 6482.761, 6517.673) (n=1082) Model 3 (ols_multiple) sacrifices a significant portion of its sample size in order to include both statistical capacity measures. While ols_multiple appears to outperform the other two models, the lower AIC/BIC partially reflects its smaller sample size, not necessarily a better model fit. Model 1 (ols_spi) provides better country coverage and sustains slightly higher explanatory power (0.7725 > 0.7614) than model 3.

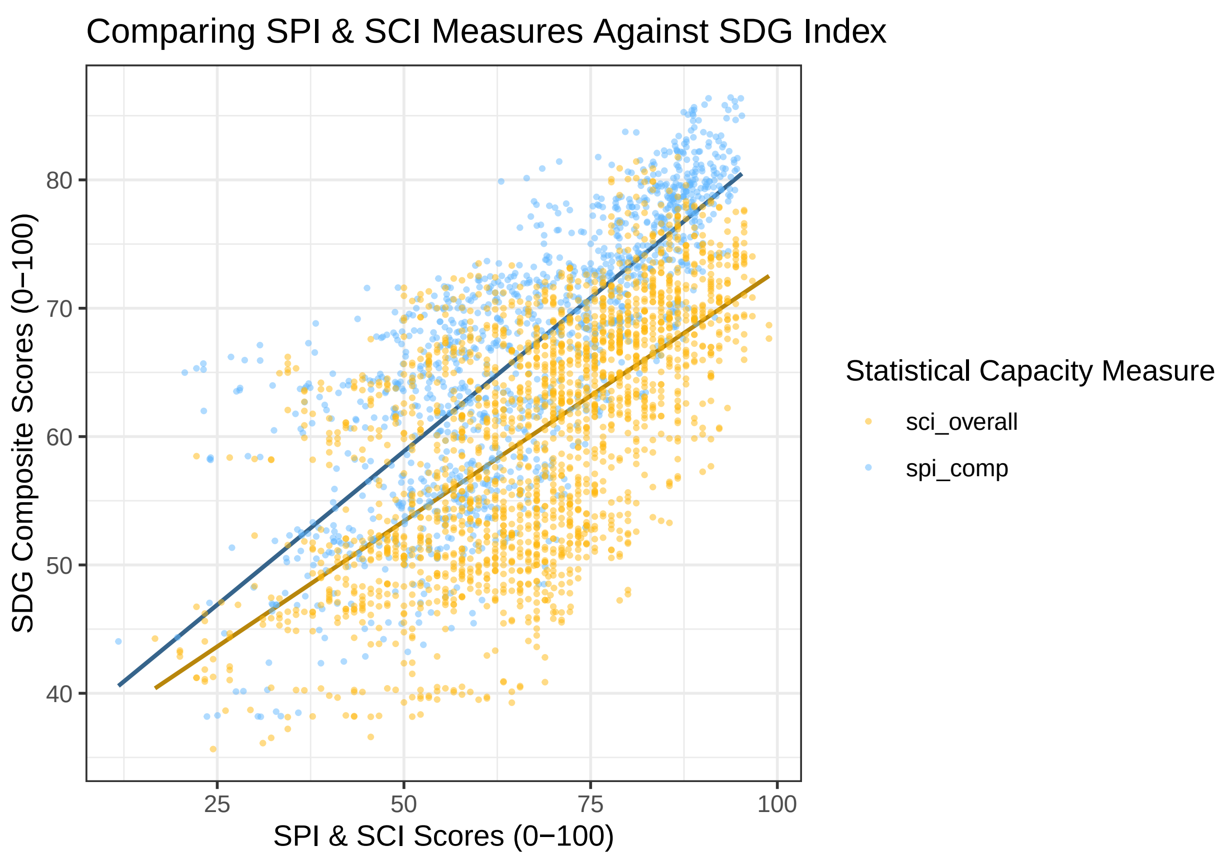
Selecting model: This analysis is specifically focused on overall statistical capacity rather than comparisons of measures. Model 1 (ols_spi) reveals a better adjusted R-squares value than that of model 2 (Adj Rsq: 0.7725 > 0.7383) and model 3 (Adj Rsq: 0.7725 > 0.7614). Model 3, containing both SPI and SCI, has greater explanatory power than model 2, but as mentioned, has a much smaller sample size, which significantly impedes results. Models 1 and 2 have significantly more country-year data points (n=1082 and n=1513, respectively) for regression analysis compared to model 3 (n=566). With all else considered, this study employs the Statistical Performance Index (SPI) as the primary measure of statistical capacity.

Visual Analysis of Fit: SCI & SPI x SDG

```r
#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
```

```r
  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 = "Comparing SPI & SCI Measures Against SDG Index",
       x = "SPI & SCI Scores (0-100)",
       y = "SDG Composite Scores (0-100)",
       color = "Statistical Capacity Measure") +  # Title for legend
  theme_bw() # Optional: adds a clean, black and white theme

Compare_fit
```

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

## Warning: Removed 2060 rows containing non-finite outside the scale range
## (`stat_smooth()`).

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

## Warning: Removed 1288 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 2060 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 1288 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```r
#make interactive
#ggplotly(Compare_fit)

# Save to specific folder
ggsave("figures/spi_sci_plot.png", Compare_fit, width = 10, height = 6)
```

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

## Warning: Removed 2060 rows containing non-finite outside the scale range
## (`stat_smooth()`).

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

## Warning: Removed 1288 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 2060 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 1288 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

The SPI regression line is expected to appear higher in terms of SDG Score compared the SCI model because SPI countries include higher-income countries. As previously mentioned, the SCI soley focuses on lower to upper-middle income countries (146 countries over 17 years).

INTERACTIONS AND SUBGROUP ANALYSIS

TEST 2: Checking for Interactions [REDO RESULTS FOR ROBUST SEs]:

- 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

```r
merged_2015 <- merged %>%
  filter(year > 2015) %>%
  mutate(regime_type_2 = as.factor(regime_type_2),
         regime_type_4 = as.factor(regime_type_4))

#interaction 1: 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 + yea
#summary(inc_lev_interaction)
coeftest(inc_lev_interaction, vcov = vcovHC(inc_lev_interaction, type = "HC1")) #Robust SE
```

```
##
## t test of coefficients:
##
##                          Estimate  Std. Error  t value  Pr(>|t|)
## (Intercept)            7.7329e+02  1.3457e+02   5.7464 1.181e-08 ***
## spi_comp               4.5536e-01  1.6984e-02  26.8121 < 2.2e-16 ***
## income_levelL          8.9189e+00  2.6519e+00   3.3632 0.0007973 ***
## income_levelLM         3.4857e+00  2.1021e+00   1.6582 0.0975684 .
## income_levelUM         2.1804e+01  1.6790e+00  12.9862 < 2.2e-16 ***
## log_gdppc              1.5321e+00  2.9655e-01   5.1665 2.834e-07 ***
## population            -3.0214e-09  7.1829e-10  -4.2064 2.807e-05 ***
## year_fct              -3.7121e-01  6.6805e-02  -5.5567 3.453e-08 ***
## spi_comp:income_levelL  -2.6155e-01  3.8181e-02  -6.8503 1.228e-11 ***
## spi_comp:income_levelLM -6.8107e-02  2.6264e-02  -2.5931 0.0096380 **
```

```
## spi_comp:income_levelUM -2.9129e-01  2.0517e-02 -14.1974 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#interaction 2: 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 + y
#summary(reg_type2_interaction)
coeftest(reg_type2_interaction, vcov = vcovHC(reg_type2_interaction, type = "HC1")) #Robust SE
```

```
##
## t test of coefficients:
##
##                           Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)              7.0587e+02  1.5638e+02  4.5139 7.052e-06 ***
## spi_comp                 2.9300e-01  1.9830e-02 14.7754 < 2.2e-16 ***
## regime_type_21           2.8606e+00  1.4990e+00  1.9083   0.05661 .
## log_gdppc                3.4372e+00  1.7514e-01 19.6251 < 2.2e-16 ***
## population              -9.8080e-10  8.1361e-10 -1.2055   0.22827
## year_fct                -3.4109e-01  7.7460e-02 -4.4034 1.170e-05 ***
## spi_comp:regime_type_21 -2.5741e-02  2.3093e-02 -1.1147   0.26524
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#interaction 3: 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 + y
                        data = merged_2015)
#summary(reg_type4_interaction)
coeftest(reg_type4_interaction, vcov = vcovHC(reg_type4_interaction, type = "HC1")) #Robust SE
```

```
##
## t test of coefficients:
##
##                           Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)              7.0868e+02  1.5894e+02  4.4588 9.091e-06 ***
## spi_comp                 2.4146e-01  4.0835e-02  5.9130 4.487e-09 ***
## regime_type_41          -2.0581e+00  2.5896e+00 -0.7948    0.4269
## regime_type_42           4.9890e-01  2.6024e+00  0.1917    0.8480
## regime_type_43           4.6605e-01  3.0937e+00  0.1506    0.8803
## log_gdppc                3.6790e+00  1.9929e-01 18.4603 < 2.2e-16 ***
## population              -8.3990e-10  8.7296e-10 -0.9621    0.3362
## year_fct                -3.4245e-01  7.8694e-02 -4.3517 1.477e-05 ***
## spi_comp:regime_type_41  5.7324e-02  4.4945e-02  1.2754    0.2024
## spi_comp:regime_type_42  3.3372e-02  4.3823e-02  0.7615    0.4465
## spi_comp:regime_type_43  1.8399e-02  4.6844e-02  0.3928    0.6946
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#interaction 4: 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 + yea
                        data = merged_2015)
#summary(reg_type_di_interaction)
coeftest(reg_type_di_interaction, vcov = vcovHC(reg_type_di_interaction, type = "HC1")) #Robust SE
```

```
##
## t test of coefficients:
##
```

```
##                       Estimate  Std. Error t value  Pr(>|t|)    
## (Intercept)         6.9059e+02  1.6023e+02  4.3099 1.783e-05 ***
## spi_comp            3.0680e-01  3.4186e-02  8.9744 < 2.2e-16 ***
## di_score            4.9966e-01  3.8571e-01  1.2954    0.1954    
## log_gdppc           3.3485e+00  1.9872e-01 16.8501 < 2.2e-16 ***
## population         -1.2355e-09  8.4665e-10 -1.4593    0.1448    
## year_fct           -3.3388e-01  7.9319e-02 -4.2093 2.775e-05 ***
## spi_comp:di_score -4.2940e-03  5.5734e-03 -0.7705    0.4412    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interaction 1: GNI Income Classification: Yes there are statistically significant interactions found from GNI classifications that affects the relationship between spi and sdgs

Interaction 2: Binary Regime Type: Yes there are statistically significant interactions (mildly, p = 0.0566) found from regime type (autocracy vs democracy) that affects the relationship between spi and sdgs.

Interaction 3: Categorical Regime type (4 options): No there are statistically significant interactions found depending on regime type (Closed autocracy, electoral autocracy, electoral democracy, liberal democracy) that affects the relationship between spi and sdgs.

Interaction 4: Continuous di_score [0-1] Regime type: No there are no statistically significant interactions found from regime type (infinate between 0-10) that affects the relationship between spi and sdgs.

Interactions DF

```r
################ in a table #################
ct <- coeftest(reg_type_di_interaction, vcov = vcovHC(reg_type_di_interaction, type = "HC1"))

# Convert to tidy dataframe
ct_tidy <- tidy(ct)

# In stargazer
stargazer(ct_tidy, type = "text", summary = FALSE, rownames = FALSE)
```

```
## 
## ==========================================================================================================
## term                    estimate              std.error          statistic           p.value        
## ----------------------------------------------------------------------------------------------------------
## (Intercept)         690.585262273173     160.233586975178   4.30986583593209  1.78317881862653e-05
## spi_comp            0.306803287842496   0.0341864594701953  8.97440953515459  1.24120911639424e-18
## di_score            0.499662681963158    0.385710543847325  1.29543433523803   0.195448287163883 
## log_gdppc           3.34847770441184     0.198721229744801   16.850125719894  1.01913190120046e-56
## population         -1.23549362160596e-09 8.4665154475312e-10 -1.45927049830898  0.144782533896456 
## year_fct            -0.333878829909611   0.0793189468336384 -4.20932000786496 2.77517531817944e-05
## spi_comp:di_score -0.00429403244132971  0.00557337054556771 -0.77045522206389  0.441199034302082 
## ----------------------------------------------------------------------------------------------------------
```

```r
# In tidy table
ct_tidy
```

```
## # A tibble: 7 x 5
##   term              estimate std.error statistic  p.value
##   <chr>                <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)        6.91e+2  1.60e+ 2     4.31  1.78e- 5
## 2 spi_comp           3.07e-1  3.42e- 2     8.97  1.24e-18
## 3 di_score           5.00e-1  3.86e- 1     1.30  1.95e- 1
```

```
## 4 log_gdppc          3.35e+0  1.99e- 1    16.9   1.02e-56
## 5 population        -1.24e-9  8.47e-10    -1.46  1.45e- 1
## 6 year_fct          -3.34e-1  7.93e- 2    -4.21  2.78e- 5
## 7 spi_comp:di_score -4.29e-3  5.57e- 3    -0.770 4.41e- 1
```

TEST 3: 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.

```r
# 1. Overall model (all countries)
overall_lm <- lm(sdg_overall ~ spi_comp + di_score + log_gdppc + population + year_fct,
                        data = merged_2015)
#summary(overall_lm)
coeftest(overall_lm, vcov = vcovHC(overall_lm, type = "HC1")) #Robust SE
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  7.0238e+02  1.5950e+02  4.4036 1.171e-05 ***
## spi_comp     2.8641e-01  1.5627e-02 18.3280 < 2.2e-16 ***
## di_score     2.1725e-01  1.1547e-01  1.8814   0.06019 .
## log_gdppc    3.3001e+00  1.8662e-01 17.6836 < 2.2e-16 ***
## population  -1.2152e-09  8.4979e-10 -1.4300   0.15302
## year_fct    -3.3890e-01  7.9041e-02 -4.2877 1.967e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 2. High income countries
high_inc_lm <- lm(sdg_overall ~ spi_comp + di_score + log_gdppc + population + year_fct,
                        data = merged_2015 %>%
                          filter(income_level == "H"))
#summary(high_inc_lm)
coeftest(high_inc_lm, vcov = vcovHC(high_inc_lm, type = "HC1")) #Robust SE
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  7.2180e+02  1.6284e+02  4.4325 1.274e-05 ***
## spi_comp     3.5556e-01  1.7029e-02 20.8795 < 2.2e-16 ***
## di_score     1.3271e+00  1.3688e-01  9.6950 < 2.2e-16 ***
## log_gdppc   -1.6264e+00  2.8557e-01 -5.6952 2.753e-08 ***
## population  -1.1668e-08  2.2696e-09 -5.1409 4.725e-07 ***
## year_fct    -3.3019e-01  8.1191e-02 -4.0668 5.983e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 3. Upper-middle income countries
upper_mid_lm <- lm(sdg_overall ~ spi_comp + di_score + log_gdppc + population + year_fct,
                         data = merged_2015 %>%
                           filter(income_level == "UM"))
#summary(upper_mid_lm)
coeftest(upper_mid_lm, vcov = vcovHC(upper_mid_lm, type = "HC1")) #Robust SE
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  3.5583e+02  2.3523e+02  1.5127  0.131510
## spi_comp     1.7431e-01  1.3929e-02 12.5140 < 2.2e-16 ***
## di_score    -5.2364e-01  1.2807e-01 -4.0888 5.694e-05 ***
## log_gdppc    1.4516e+00  7.1010e-01  2.0442  0.041887 *
## population  -1.3491e-09  5.0903e-10 -2.6503  0.008506 **
## year_fct    -1.5227e-01  1.1719e-01 -1.2994  0.194898
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 4. Lower-middle income countries
lower_mid_lm <- lm(sdg_overall ~ spi_comp + di_score + log_gdppc + population + year_fct,
                         data = merged_2015 %>%
                           filter(income_level == "LM"))
#summary(lower_mid_lm)
coeftest(lower_mid_lm, vcov = vcovHC(lower_mid_lm, type = "HC1")) #Robust SE
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  8.9764e+02  2.9958e+02  2.9964  0.002965 **
## spi_comp     3.8222e-01  2.6271e-02 14.5492 < 2.2e-16 ***
## di_score    -3.9289e-01  1.8738e-01 -2.0967  0.036871 *
## log_gdppc    5.0617e+00  7.5035e-01  6.7458 8.080e-11 ***
## population  -3.2231e-09  7.1371e-10 -4.5160 9.134e-06 ***
## year_fct    -4.4362e-01  1.4865e-01 -2.9844  0.003080 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 5. Low income countries
low_inc_lm <- lm(sdg_overall ~ spi_comp + di_score + log_gdppc + population + year_fct,
                       data = merged_2015 %>%
                         filter(income_level == "L"))
#summary(low_inc_lm)
coeftest(low_inc_lm, vcov = vcovHC(low_inc_lm, type = "HC1")) #Robust SE
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  3.5341e+02  3.8840e+02  0.9099 0.3642116
## spi_comp     1.6965e-01  4.4132e-02  3.8442 0.0001731 ***
## di_score    -3.4255e-03  3.1974e-01 -0.0107 0.9914654
## log_gdppc    4.2132e+00  9.6565e-01  4.3630 2.269e-05 ***
## population  -3.2493e-08  9.0169e-09 -3.6036 0.0004165 ***
## year_fct    -1.6647e-01  1.9305e-01 -0.8623 0.3898013
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

GNI Classifications DF

```r
# 1. Overall model
overall_ct <- coeftest(overall_lm, vcov = vcovHC(overall_lm, type = "HC1"))
overall_df <- data.frame(
  model = "overall_lm",
  term = rownames(overall_ct),
  estimate = overall_ct[, "Estimate"],
  std.error = overall_ct[, "Std. Error"],
  t.statistic = overall_ct[, "t value"],
  p.value = overall_ct[, "Pr(>|t|)"],
  n = nobs(overall_lm)
)

# 2. High income countries
high_inc_ct <- coeftest(high_inc_lm, vcov = vcovHC(high_inc_lm, type = "HC1"))
high_inc_df <- data.frame(
  model = "high_inc_lm",
  term = rownames(high_inc_ct),
  estimate = high_inc_ct[, "Estimate"],
  std.error = high_inc_ct[, "Std. Error"],
  t.statistic = high_inc_ct[, "t value"],
  p.value = high_inc_ct[, "Pr(>|t|)"],
  n = nobs(high_inc_lm)
)

# 3. Upper-middle income countries
upper_mid_ct <- coeftest(upper_mid_lm, vcov = vcovHC(upper_mid_lm, type = "HC1"))
upper_mid_df <- data.frame(
  model = "upper_mid_lm",
  term = rownames(upper_mid_ct),
  estimate = upper_mid_ct[, "Estimate"],
  std.error = upper_mid_ct[, "Std. Error"],
  t.statistic = upper_mid_ct[, "t value"],
  p.value = upper_mid_ct[, "Pr(>|t|)"],
  n = nobs(upper_mid_lm)
)

# 4. Lower-middle income countries
lower_mid_ct <- coeftest(lower_mid_lm, vcov = vcovHC(lower_mid_lm, type = "HC1"))
lower_mid_df <- data.frame(
  model = "lower_mid_lm",
  term = rownames(lower_mid_ct),
  estimate = lower_mid_ct[, "Estimate"],
  std.error = lower_mid_ct[, "Std. Error"],
  t.statistic = lower_mid_ct[, "t value"],
  p.value = lower_mid_ct[, "Pr(>|t|)"],
  n = nobs(lower_mid_lm)
)

# 5. Low income countries
low_inc_ct <- coeftest(low_inc_lm, vcov = vcovHC(low_inc_lm, type = "HC1"))
low_inc_df <- data.frame(
  model = "low_inc_lm",
```

```r
  term = rownames(low_inc_ct),
  estimate = low_inc_ct[, "Estimate"],
  std.error = low_inc_ct[, "Std. Error"],
  t.statistic = low_inc_ct[, "t value"],
  p.value = low_inc_ct[, "Pr(>|t|)"],
  n = nobs(low_inc_lm)
)

# Combine all results
gni_classes_ols <- bind_rows(
  overall_df,
  high_inc_df,
  upper_mid_df,
  lower_mid_df,
  low_inc_df
)

attr(gni_classes_ols $std.error, "label") <- "Robust Std. Errors Adjusted"
attr(gni_classes_ols $t.statistic, "label") <- "Robust Std. Errors Adjusted"
attr(gni_classes_ols $p.value, "label") <- "Robust Std. Errors Adjusted"

gni_classes_ols
```

```
##                         model        term      estimate    std.error
## (Intercept)...1    overall_lm (Intercept)  7.023774e+02 1.595014e+02
## spi_comp...2       overall_lm    spi_comp  2.864147e-01 1.562713e-02
## di_score...3       overall_lm    di_score  2.172487e-01 1.154748e-01
## log_gdppc...4      overall_lm   log_gdppc  3.300063e+00 1.866166e-01
## population...5     overall_lm  population -1.215172e-09 8.497912e-10
## year_fct...6       overall_lm    year_fct -3.389012e-01 7.904091e-02
## (Intercept)...7   high_inc_lm (Intercept)  7.217983e+02 1.628429e+02
## spi_comp...8      high_inc_lm    spi_comp  3.555640e-01 1.702933e-02
## di_score...9      high_inc_lm    di_score  1.327082e+00 1.368838e-01
## log_gdppc...10    high_inc_lm   log_gdppc -1.626376e+00 2.855707e-01
## population...11   high_inc_lm  population -1.166780e-08 2.269623e-09
## year_fct...12     high_inc_lm    year_fct -3.301853e-01 8.119132e-02
## (Intercept)...13 upper_mid_lm (Intercept)  3.558261e+02 2.352317e+02
## spi_comp...14    upper_mid_lm    spi_comp  1.743138e-01 1.392949e-02
## di_score...15    upper_mid_lm    di_score -5.236355e-01 1.280673e-01
## log_gdppc...16   upper_mid_lm   log_gdppc  1.451560e+00 7.101024e-01
## population...17  upper_mid_lm  population -1.349085e-09 5.090321e-10
## year_fct...18    upper_mid_lm    year_fct -1.522710e-01 1.171875e-01
## (Intercept)...19 lower_mid_lm (Intercept)  8.976413e+02 2.995778e+02
## spi_comp...20    lower_mid_lm    spi_comp  3.822174e-01 2.627076e-02
## di_score...21    lower_mid_lm    di_score -3.928911e-01 1.873820e-01
## log_gdppc...22   lower_mid_lm   log_gdppc  5.061675e+00 7.503496e-01
## population...23  lower_mid_lm  population -3.223103e-09 7.137094e-10
## year_fct...24    lower_mid_lm    year_fct -4.436210e-01 1.486456e-01
## (Intercept)...25   low_inc_lm (Intercept)  3.534141e+02 3.884041e+02
## spi_comp...26      low_inc_lm    spi_comp  1.696516e-01 4.413233e-02
## di_score...27      low_inc_lm    di_score -3.425456e-03 3.197427e-01
## log_gdppc...28     low_inc_lm   log_gdppc  4.213167e+00 9.656528e-01
## population...29    low_inc_lm  population -3.249348e-08 9.016871e-09
## year_fct...30      low_inc_lm    year_fct -1.664658e-01 1.930535e-01
```

```
##                   t.statistic      p.value    n
## (Intercept)...1   4.40358216 1.170908e-05 1083
## spi_comp...2     18.32804413 1.609530e-65 1083
## di_score...3      1.88135145 6.019368e-02 1083
## log_gdppc...4    17.68364784 1.234510e-61 1083
## population...5   -1.42996530 1.530170e-01 1083
## year_fct...6     -4.28766768 1.967494e-05 1083
## (Intercept)...7   4.43248218 1.274354e-05  332
## spi_comp...8     20.87950078 4.627872e-62  332
## di_score...9      9.69495630 1.075841e-19  332
## log_gdppc...10   -5.69517726 2.753203e-08  332
## population...11  -5.14085278 4.724826e-07  332
## year_fct...12    -4.06675688 5.982931e-05  332
## (Intercept)...13  1.51266198 1.315097e-01  282
## spi_comp...14    12.51400779 9.214843e-29  282
## di_score...15    -4.08875191 5.693698e-05  282
## log_gdppc...16    2.04415643 4.188704e-02  282
## population...17  -2.65029387 8.506373e-03  282
## year_fct...18    -1.29937955 1.948980e-01  282
## (Intercept)...19  2.99635415 2.965149e-03  300
## spi_comp...20    14.54915767 1.706294e-36  300
## di_score...21    -2.09673822 3.687102e-02  300
## log_gdppc...22    6.74575474 8.079920e-11  300
## population...23  -4.51598783 9.133549e-06  300
## year_fct...24    -2.98441946 3.079706e-03  300
## (Intercept)...25  0.90991353 3.642116e-01  169
## spi_comp...26     3.84415690 1.731056e-04  169
## di_score...27    -0.01071316 9.914654e-01  169
## log_gdppc...28    4.36302413 2.268567e-05  169
## population...29  -3.60363108 4.164809e-04  169
## year_fct...30    -0.86227810 3.898013e-01  169
```

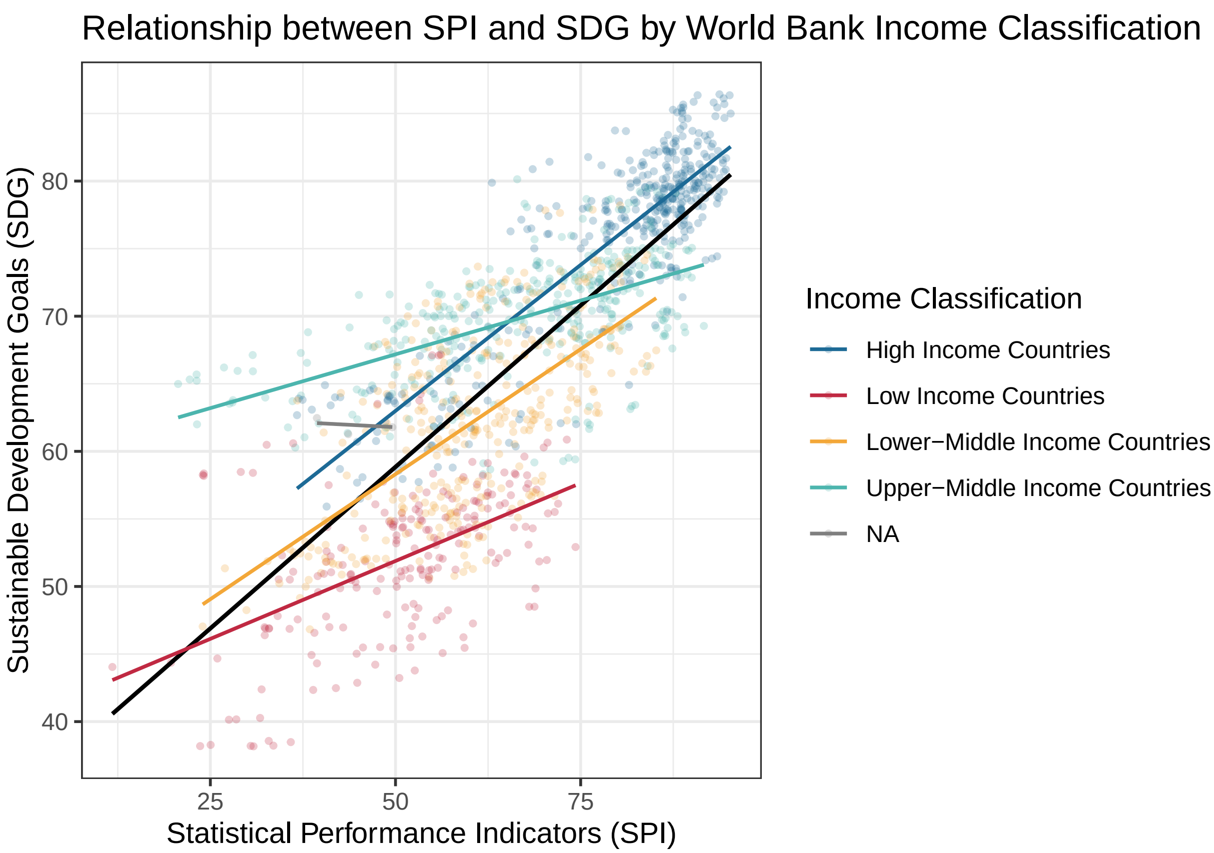
```
#interactive table for side access
#datatable(gni_classes_ols, caption = "Regression Results, by GNI Country Classifications")

# write to directory
write.csv(gni_classes_ols, "output_CSVs/gni_classes_ols.csv")
```

##Visualizing Slopes: plotting multiple regression - by subgroup

```
viz_gni_class <- ggplot(data = merged_2015, 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(
```

```r
    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 44 rows containing non-finite outside the scale range
## (`stat_smooth()`).

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

## Warning: Removed 44 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 44 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```r
#ggplotly(viz_gni_class)

# Save to specific folder
```

```
ggsave("figures/gni_subgroups_ols.png", viz_gni_class, width = 10, height = 6)
```

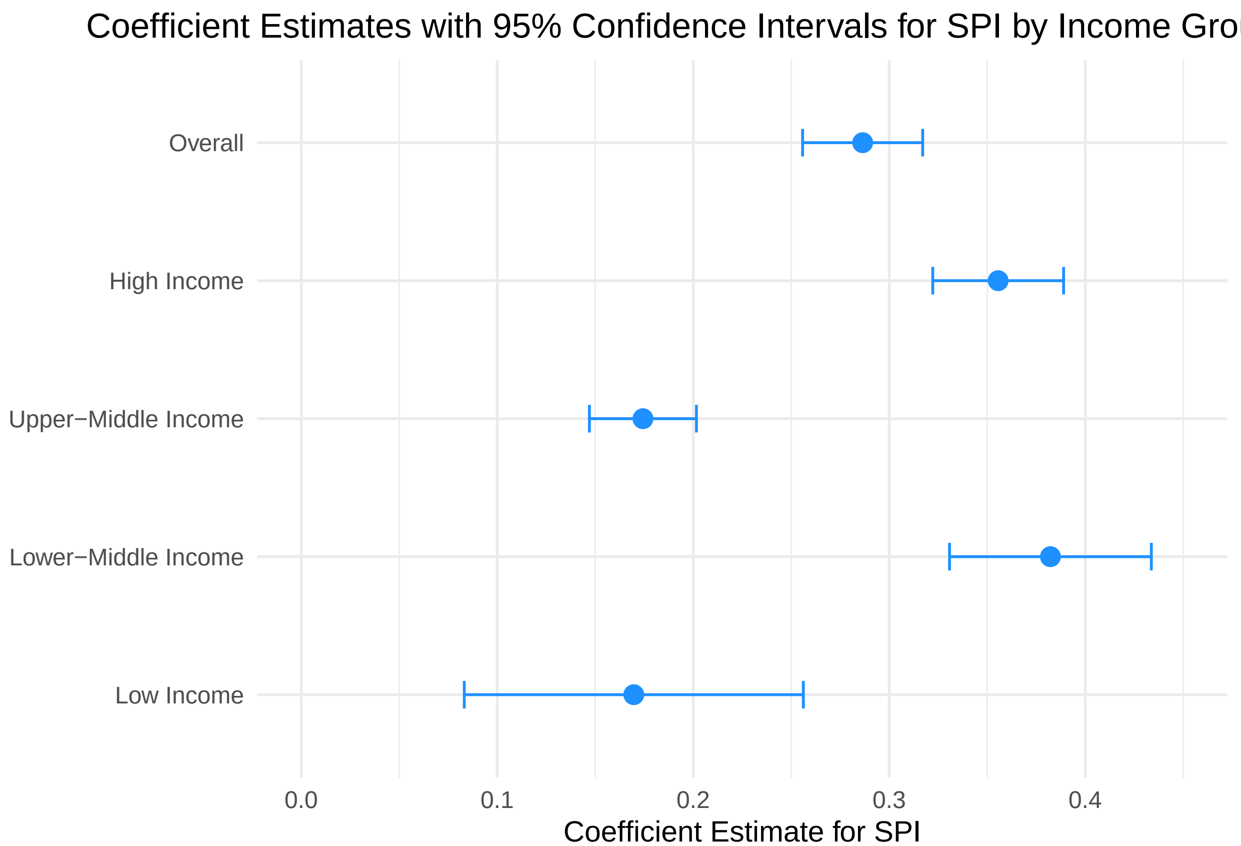
```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 44 rows containing non-finite outside the scale range
## (`stat_smooth()`).

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

## Warning: Removed 44 rows containing non-finite outside the scale range
## (`stat_smooth()`).
## Removed 44 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

[more results TBD]

Coefficient & Interval Plot

```
# New fd with SPI coefficients data
spi_plot_data <- data.frame(
  model = c("overall_lm", "high_inc_lm", "upper_mid_lm", "lower_mid_lm", "low_inc_lm"),
  estimate = c(0.286414727883271, 0.355563975838364, 0.174313752024353, 0.382217438772999, 0.16965161402
  std.error = c(0.0156271299809294, 0.0170293332032076, 0.0139294904502819, 0.0262707606446246, 0.04413
)

# Calculate confidence intervals
spi_plot_data <- spi_plot_data %>%
  mutate(
    CI_lower = estimate - 1.96 * std.error,
    CI_upper = estimate + 1.96 * std.error
  )

# Set model order
model_order <- c("low_inc_lm", "lower_mid_lm", "upper_mid_lm", "high_inc_lm", "overall_lm")
spi_plot_data$model <- factor(spi_plot_data$model, levels = model_order)

# Create the coefficient plot
coef_inter_spi_plot <- ggplot(spi_plot_data, aes(x = estimate, y = model)) +
  geom_point(size = 3, color = "dodgerblue") +
  geom_errorbarh(aes(xmin = CI_lower, xmax = CI_upper),
                 height = 0.2,
                 color = "dodgerblue") +
  labs(
    title = "Coefficient Estimates with 95% Confidence Intervals for SPI by Income Group Models",
    x = "Coefficient Estimate for SPI",
    y = NULL
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_x_continuous(limits = c(0, 0.45)) +
  scale_y_discrete(labels = c("low_inc_lm" = "Low Income",
                              "lower_mid_lm" = "Lower-Middle Income",
                              "upper_mid_lm" = "Upper-Middle Income",
                              "high_inc_lm" = "High Income",
                              "overall_lm" = "Overall"))
```

```r
coef_inter_spi_plot
```




```r
ggsave("figures/coef_inter_spi_plot.png", coef_inter_spi_plot, width = 9, height = 5)
```