Lab3

Sevastian Sanchez

2024-12-11

Questions:

1. Create a multivariate time series; perform any interpolations.
2. Graph the relationships between X and Y. Explain how you think Y should relate to your key Xs.
3. Run a simple time series regression, with one X and no trend. Interpret it. ## 4. Run a time series regression with one X and trend. Interpret it. Perform autocorrelation diagnostics. Explain what you found.
5. Consider running a time series regression with many Xs and trend. Interpret that. Check VIF. ## 6. Run a first differenced time series regression. Interpret that.
7. Check your variables for unit roots. Do some tests. Interpret them. ## 8. Perform an Automatic ARIMA on the residuals from one of your earlier models. Tell me what it says. ## 9. Run an ARIMA that follows from Step 8. Interpret that, too.

```
# packages installed via console load packages
library(QMSS)
```

```
## Loading required package: lme4

## Loading required package: Matrix

## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

## Loading required package: MASS

## Loading required package: plm

## Loading required package: plyr

## Loading required package: rdd

## Loading required package: sandwich

## Loading required package: AER

## Loading required package: car

## Loading required package: carData

## Loading required package: survival

## Loading required package: Formula

## Loading required package: VGAM
```

```
## Loading required package: stats4
## Loading required package: splines
##
## Attaching package: 'VGAM'
## The following object is masked from 'package:AER':
##
##     tobit
## The following object is masked from 'package:car':
##
##     logit
## The following object is masked from 'package:plm':
##
##     has.intercept
## The following object is masked from 'package:lmtest':
##
##     lrtest
```

```r
library(reshape2)
library(ggplot2)
library(plyr)
library(car)
library(fUnitRoots)
library(lmtest)
library(readxl)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
## The following object is masked from 'package:reshape2':
##
##     smiths
## The following objects are masked from 'package:Matrix':
##
##     expand, pack, unpack
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:car':
##
##     recode
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
## The following objects are masked from 'package:plm':
##
##     between, lag, lead
```

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

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following objects are masked from 'package:reshape2':
##
##     dcast, melt

## The following object is masked from 'package:plm':
##
##     between

## The following objects are masked from 'package:zoo':
##
##     yearmon, yearqtr
```

```r
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(knitr)
library(formatR)
```

QUESTION: How does countries' SDG performance (measured by SDG index) react to changes in statistical capacity (measured by the Statistical Capacity Index (SCI)) over time.

1. Create a multivariate time series, perform any interpolations.

I need the dataset to contain variables representing sdg performance and drivers such as statistical capacity, GDP and political factors. I need these variables represented across time, specifically in annual intervals from 2000-2023.

```r
# set working directory
setwd("~/Desktop/OneDrive/OneDrive/Datasets")

# load vdem data
vdem <- read.csv("V-Dem_Data.csv")
vdem <- select(vdem, "country_name", "country_text_id", "year",
    "v2x_accountability", "v2x_libdem", "v2xca_academ", "e_gdppc",
    "e_wb_pop")
vdem <- vdem[vdem$year > 1999, ]
vdem <- vdem %>%
    rename(country_code = "country_text_id")
vdem$v2xca_academ <- vdem$v2xca_academ * 100
vdem$v2x_libdem <- vdem$v2x_libdem * 100

# load SPI data
spi <- read_excel("~/Desktop/OneDrive/OneDrive/Datasets/SPI_index.xlsx",
    na = "NA")
spi <- select(spi, "country_name", "country_code", "year", "spi_overall",
    "spi_p1", "spi_p2", "spi_p3", "spi_p4", "spi_p5")
summary(spi)
```

```
##  country_name       country_code            year       spi_overall
##  Length:4340        Length:4340        Min.   :2004   Min.   :11.77
##  Class :character   Class :character   1st Qu.:2009   1st Qu.:52.84
##  Mode  :character   Mode  :character   Median :2014   Median :64.28
##                                        Mean   :2014   Mean   :64.95
##                                        3rd Qu.:2018   3rd Qu.:80.20
##                                        Max.   :2023   Max.   :95.26
##                                                       NA's   :2915
##      spi_p1           spi_p2              spi_p3           spi_p4
##  Min.   :  0.00   Min.   :  0.3333   Min.   : 4.894   Min.   : 0.00
##  1st Qu.: 30.00   1st Qu.: 56.1833   1st Qu.:45.506   1st Qu.:36.88
##  Median : 40.00   Median : 64.0000   Median :58.025   Median :52.82
##  Mean   : 50.75   Mean   : 64.7815   Mean   :55.213   Mean   :51.89
##  3rd Qu.: 80.00   3rd Qu.: 86.4667   3rd Qu.:68.431   3rd Qu.:68.63
##  Max.   :100.00   Max.   :100.0000   Max.   :94.312   Max.   :94.17
##                   NA's   :2904       NA's   :255      NA's   :2780
##      spi_p5
##  Min.   :  0.00
##  1st Qu.: 30.00
##  Median : 50.00
##  Mean   : 54.94
##  3rd Qu.: 80.00
##  Max.   :100.00
##  NA's   :2821
```

```r
str(spi)
```

```
## tibble [4,340 x 9] (S3: tbl_df/tbl/data.frame)
##  $ country_name: chr [1:4340] "Denmark" "Finland" "Poland" "Sweden" ...
##  $ country_code: chr [1:4340] "DNK" "FIN" "POL" "SWE" ...
##  $ year        : num [1:4340] 2023 2023 2023 2023 2023 ...
##  $ spi_overall : num [1:4340] 95.3 95.1 94.7 94.4 94.3 ...
##  $ spi_p1      : num [1:4340] 100 100 100 100 100 100 100 100 100 100 ...
```

```
##  $ spi_p2      : num [1:4340] 98.5 96.4 97.3 96 91.2 ...
##  $ spi_p3      : num [1:4340] 90.7 91 84.5 90.6 92.6 ...
##  $ spi_p4      : num [1:4340] 87.1 88.2 91.4 85.5 87.8 ...
##  $ spi_p5      : num [1:4340] 100 100 100 100 100 100 100 100 100 100 ...
```

```
# load SDG data
sdg <- read_excel("~/Desktop/OneDrive/OneDrive/Datasets/SDR2024-data.xlsx",
    sheet = "Backdated SDG Index")
sdg <- select(sdg, "country_code", "country_name", "year", "sdg_overall",
    "goal1", "goal2", "goal3", "goal4", "goal5", "goal6", "goal7",
    "goal8", "goal9", "goal10", "goal11", "goal12", "goal3",
    "goal14", "goal15", "goal16", "goal17")
```

Use *na.rm = TRUE* in calculations to handle missing values

Merging & subsetting (case study: Spain)

```
# merge spi, sdg and vdem data
merged <- merge(sdg, vdem, by = c("country_code", "year"))
merged <- merge(merged, spi, by = c("country_code", "year"))

# data type changes
merged$year <- as.integer(merged$year)
merged <- merged %>%
    mutate_at(c("spi_overall", "spi_p1", "spi_p2", "spi_p3",
        "spi_p4", "spi_p5"), as.numeric)

# subset for Spain
merged_ESP <- merged[merged$country_code == "ESP", ]
```

setting up timeseries

```
# variables for time series analysis
keep.vars <- c("year", "sdg_overall", "spi_overall", "v2x_accountability",
    "v2x_libdem", "v2xca_academ", "e_gdppc", "e_wb_pop")

meltMyTS <- function(mv.ts.object, time.var, keep.vars) {
    # mv.ts.object = a multivariate ts object keep.vars =
    # character vector with names of variables to keep
    # time.var = character string naming the time variable
    require(reshape2)

    if (missing(keep.vars)) {
        melt.dat <- data.frame(mv.ts.object)
    } else {
        if (!(time.var %in% keep.vars)) {
            keep.vars <- c(keep.vars, time.var)
        }
        melt.dat <- data.frame(mv.ts.object)[, keep.vars]
    }
    melt.dat <- melt(melt.dat, id.vars = time.var)
    colnames(melt.dat)[which(colnames(melt.dat) == time.var)] <- "time"
```

```r
    return(melt.dat)
}
```

2. Graph the relationships between X and Y. Explain how you think Y should relate to your key Xs.

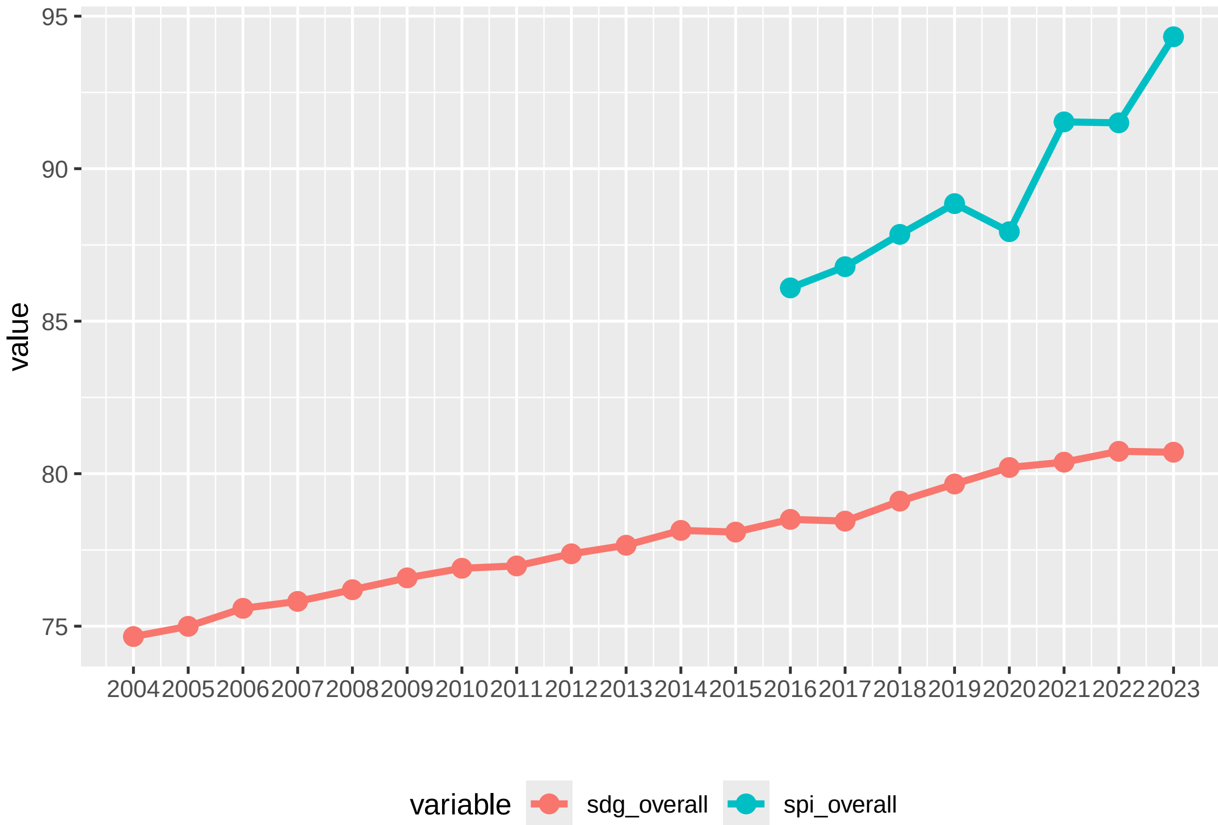
```r
# ESP time series
plot.data <- meltMyTS(mv.ts.object = merged_ESP, time.var = "year",
    keep.vars = keep.vars)
```

```
## Warning in melt.default(melt.dat, id.vars = time.var): The melt generic in
## data.table has been passed a data.frame and will attempt to redirect to the
## relevant reshape2 method; please note that reshape2 is superseded and is no
## longer actively developed, and this redirection is now deprecated. To continue
## using melt methods from reshape2 while both libraries are attached, e.g.
## melt.list, you can prepend the namespace, i.e. reshape2::melt(melt.dat). In the
## next version, this warning will become an error.
```

```r
# Use ggMyTS to plot 2 variables Y=sdg_overall,
# x=spi_overall
(plot.data1 <- ggMyTS(df = plot.data, varlist = c("sdg_overall",
    "spi_overall")))
```

```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

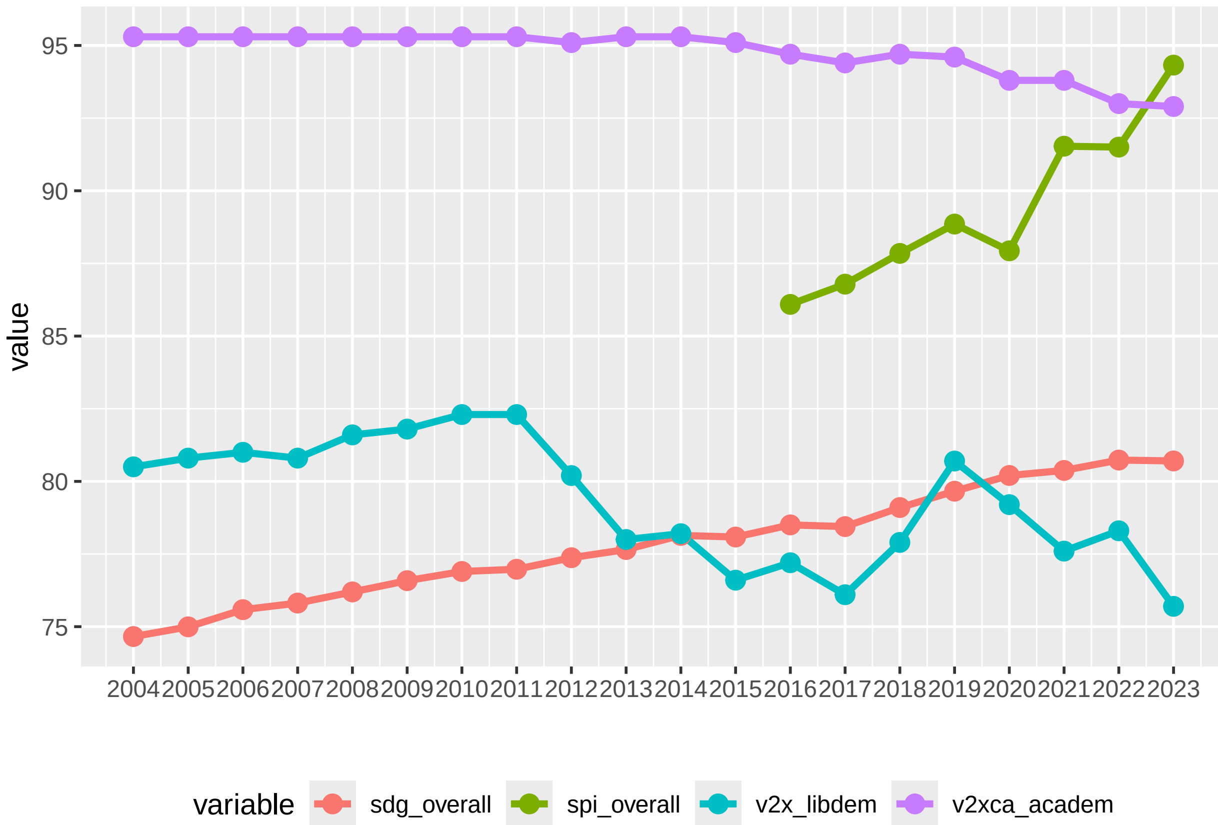


#Explain how you think Y should relate to your Xs Answer: I suspect that higher SPI and GDP likely contribute to better SDG outcomes due to enhanced resources and institutional capacity. Academic Freedom and Government Accountability may have indirect or lagged effects.

```
# Use ggMyTS to plot 4 main variables together
(plot.data2 <- ggMyTS(df = plot.data, varlist = c("sdg_overall",
    "spi_overall", "v2x_libdem", "v2xca_academ")))
```

```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Plotting varible time series' side by side

```
ggMyTS <- function(df, varlist, line = TRUE, point = TRUE, pointsize = 3,
    linewidth = 1.25, ...) {
    require(ggplot2)
    # varlist = character vector with names of variables to
    # use
    if (missing(varlist)) {
        gg <- ggplot(df, aes(time, value, colour = variable))
    } else {
        include <- with(df, variable %in% varlist)
        gg <- ggplot(df[include, ], aes(time, value, colour = variable))
    }
    if (line == FALSE & point == FALSE) {
        stop("At least one of 'line' or 'point' must be TRUE")
    } else {
        if (line == TRUE)
            gg <- gg + geom_line(size = linewidth, aes(color = variable),
                ...)
        if (point == TRUE)
            gg <- gg + geom_point(size = pointsize, aes(color = variable),
                ...)
    }

    gg + xlab("") + theme(legend.position = "bottom") + scale_x_continuous(breaks = min(df$time):max(df$
}
```

```r
# first we can make a plot for each of the variables of
# interest
sdg_plot <- ggMyTS(plot.data, "sdg_overall", color = "blue") +
    ylab("SDG Performance")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
spi_plot <- ggMyTS(plot.data, "spi_overall", color = "purple2") +
    ylab("Statistical Capacity")
gdp_plot <- ggMyTS(plot.data, "e_gdppc", color = "forestgreen") +
    ylab("GDP per capita")
academ_plot <- ggMyTS(plot.data, "v2xca_academ", color = "red2") +
    ylab("Academic Freedom")
libdem_plot <- ggMyTS(plot.data, "v2x_libdem", color = "orange") +
    ylab("'Liberal Democracy' Level")

x_axis <- scale_x_continuous(breaks = seq(2000, 2023, by = 1))
sdg_plot <- sdg_plot + x_axis
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```r
spi_plot <- spi_plot + x_axis
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```r
gdp_plot <- gdp_plot + x_axis
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```r
libdem_plot <- libdem_plot + x_axis
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```r
academ_plot <- academ_plot + x_axis
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```r
# now we can use grid.arrange to plot variables side by
# side
grid.arrange(sdg_plot, spi_plot, gdp_plot, academ_plot)
```

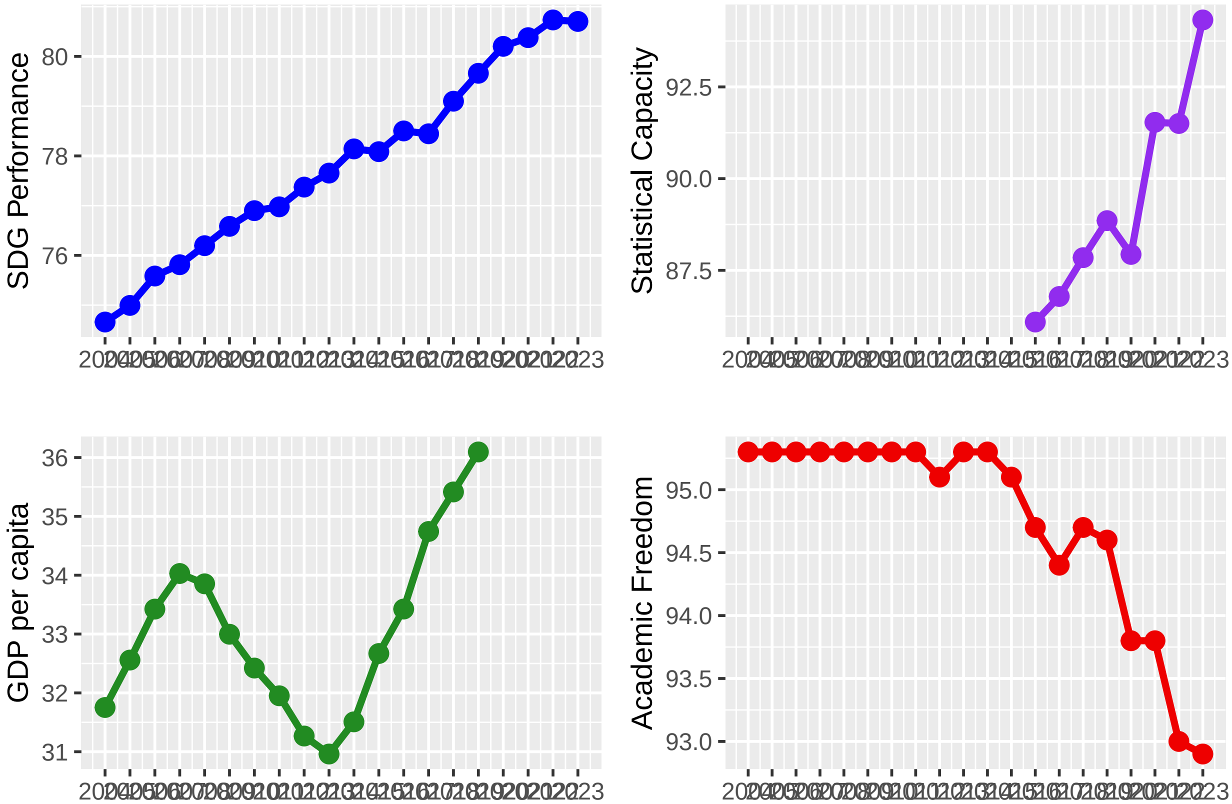
```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 4 rows containing missing values or values outside the scale range
```

```
## (`geom_point()`).
```



Plotting multi-varible time series' side by side

```r
# plotting two variables in a timeseries (SDG + var)
sdg_spi_plot <- ggMyTS(plot.data, c("sdg_overall", "spi_overall")) +
    ylab("SDG Performance and Statistical Capacity")
sdg_gdp_plot <- ggMyTS(plot.data, c("sdg_overall", "e_gdppc")) +
    ylab("SDG Performance and GDP per capita")
sdg_academ_plot <- ggMyTS(plot.data, c("sdg_overall", "v2xca_academ")) +
    ylab("SDG Performance and Academic Freedom")
sdg_libdem_plot <- ggMyTS(plot.data, c("sdg_overall", "v2x_libdem")) +
    ylab("SDG Performance and 'Liberal Democracy' Level")

# instead of using the very wordy
# 'scale_x_continuous(breaks = seq(1972,1992, by = 4))' for
# each of the plots, we can assign it to an object with a
# shorter

# name and use it like this:
x_axis <- scale_x_continuous(breaks = seq(2000, 2023, by = 1))
sdg_spi_plot <- sdg_spi_plot + x_axis
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```r
sdg_gdp_plot <- sdg_gdp_plot + x_axis
```

```
## Scale for x is already present.
```

```
## Adding another scale for x, which will replace the existing scale.
```

```r
sdg_academ_plot <- sdg_academ_plot + x_axis
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```r
sdg_libdem_plot <- sdg_libdem_plot + x_axis
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

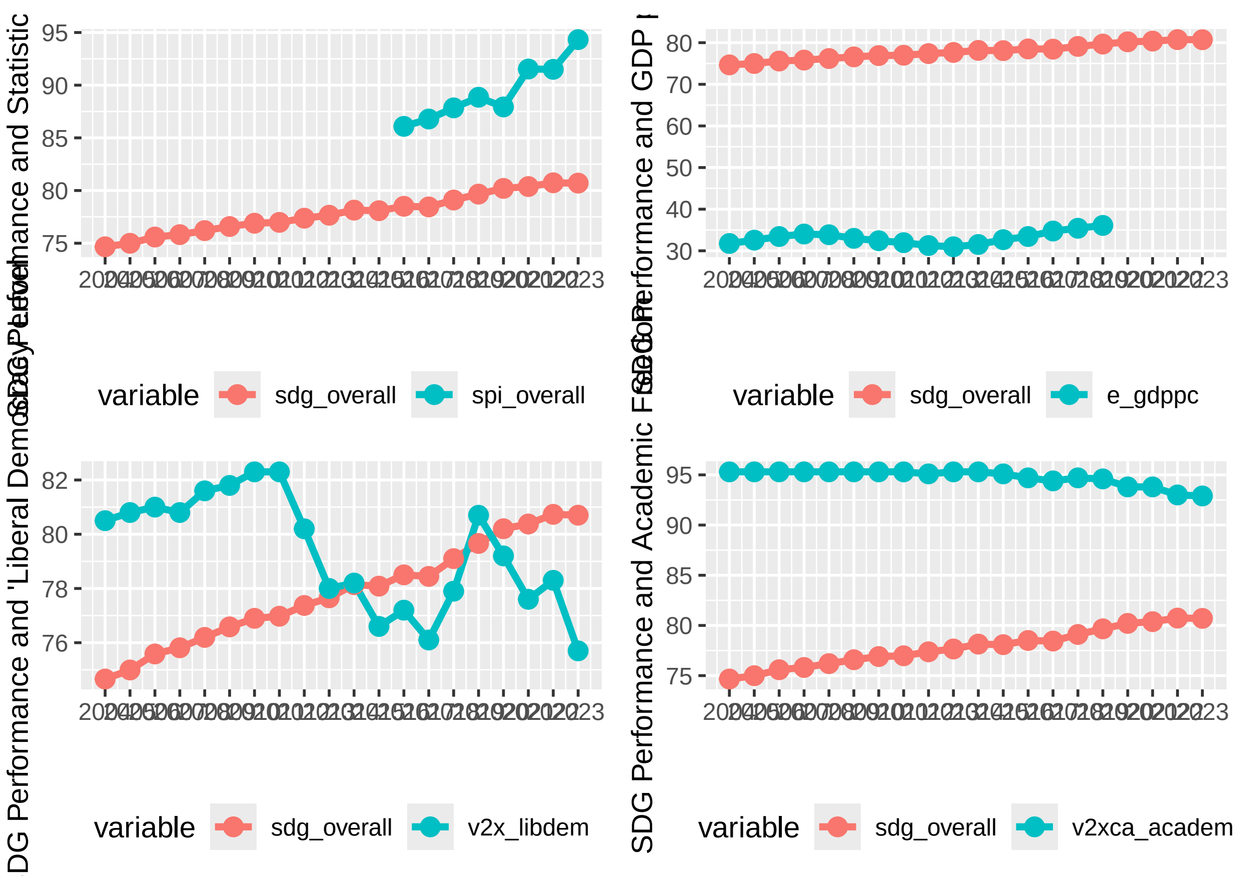
```r
# now we can use grid.arrange to plot graphs of more than 1
# variable side by side
grid.arrange(sdg_spi_plot, sdg_gdp_plot, sdg_libdem_plot, sdg_academ_plot)
```

```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Answer: Upon visualizing the relationships between my Xs and Y, we see that GDP and SPI for a similar direction with SGD performance, which suggestes a potential correlation that warrants further exploration.

3. Run a simple time series regression, with one X and no trend. Interpret it.

```r
# simplest regression (all data)
lm.sdg_spi <- lm(sdg_overall ~ spi_overall, data = merged)
summary(lm.sdg_spi)
```

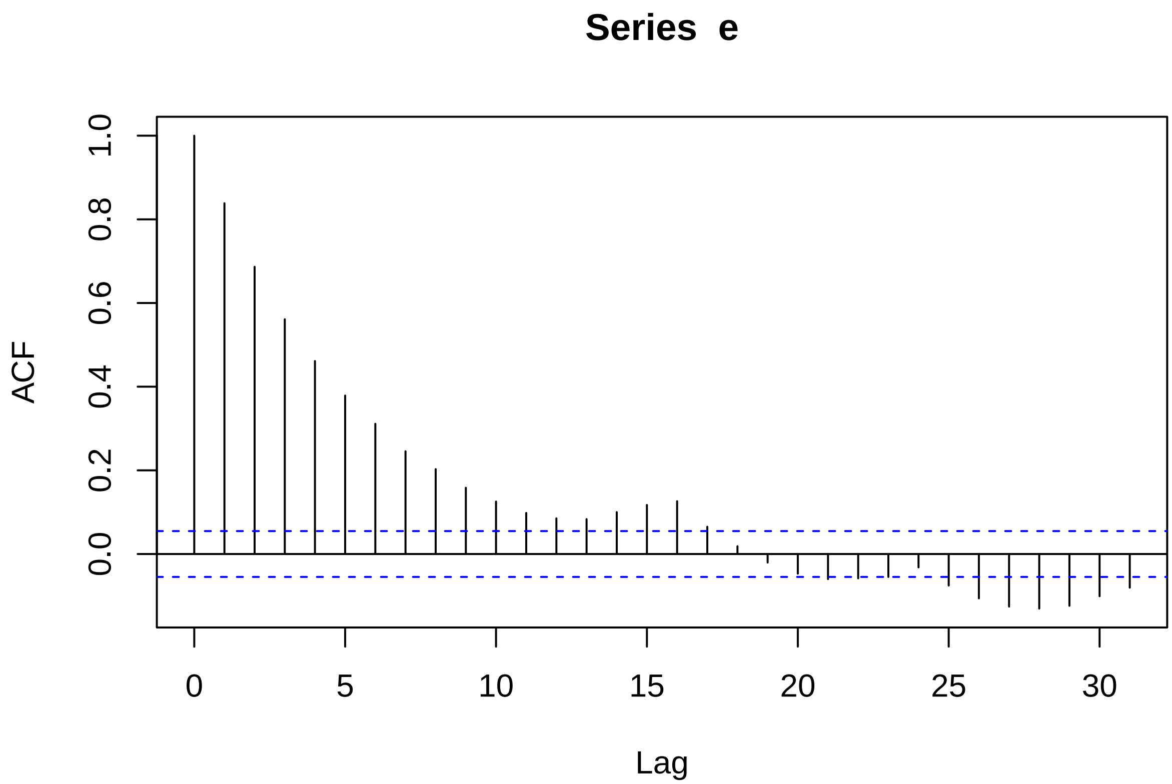
```
##
## Call:
## lm(formula = sdg_overall ~ spi_overall, data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.2429  -4.3999   0.5628   4.4048  20.5287
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.46337    0.72793   47.34   <2e-16 ***
## spi_overall  0.48400    0.01055   45.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.334 on 1278 degrees of freedom
##   (1993 observations deleted due to missingness)
## Multiple R-squared:  0.6222, Adjusted R-squared:  0.6219
## F-statistic:  2105 on 1 and 1278 DF,  p-value: < 2.2e-16
```

The results show a statistically significant positive relationship between SPI and SDG performance (coefficient = 0.48400, $p < 0.001$). This indicates that for every one-point increase in the statistical capacity, there is an associated 0.484 point increase in SDG performance. The model explains about 62% of the variance in SDG performance (R-squared = 0.6222).

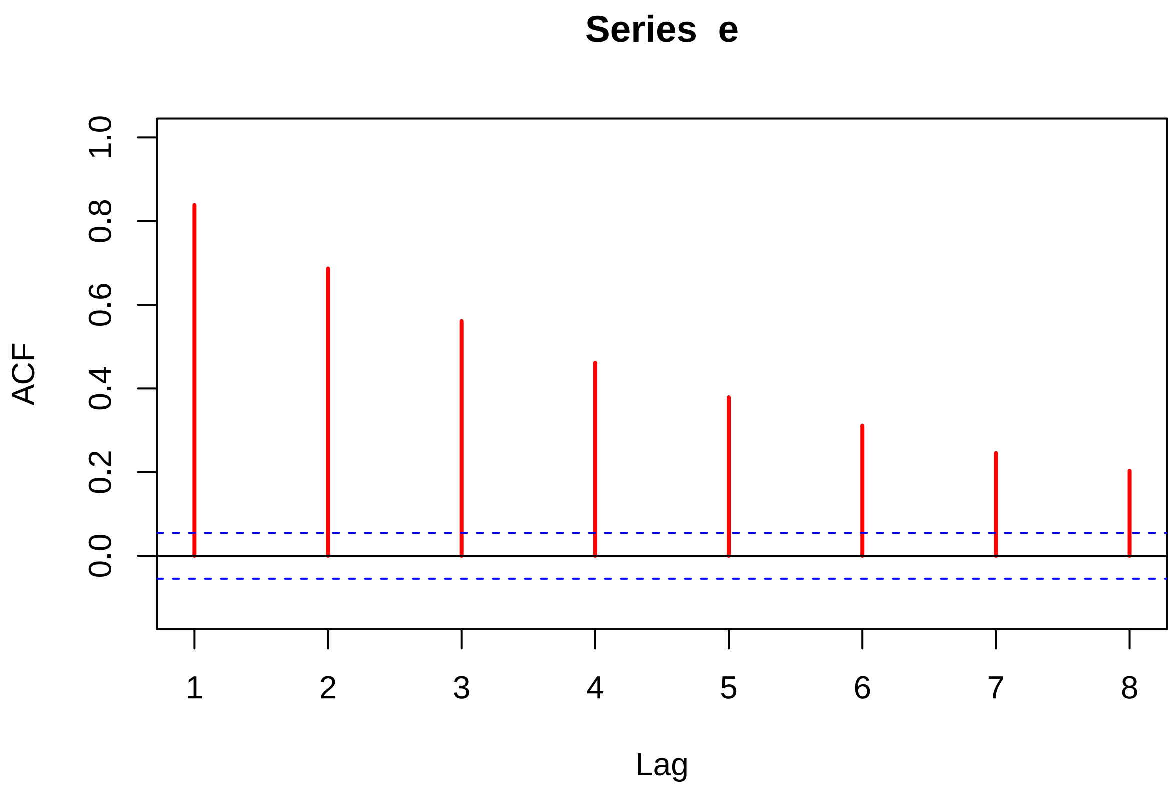
```r
# test for heteroskedasticity
bptest(lm.sdg_spi)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm.sdg_spi
## BP = 156.68, df = 1, p-value < 2.2e-16
```

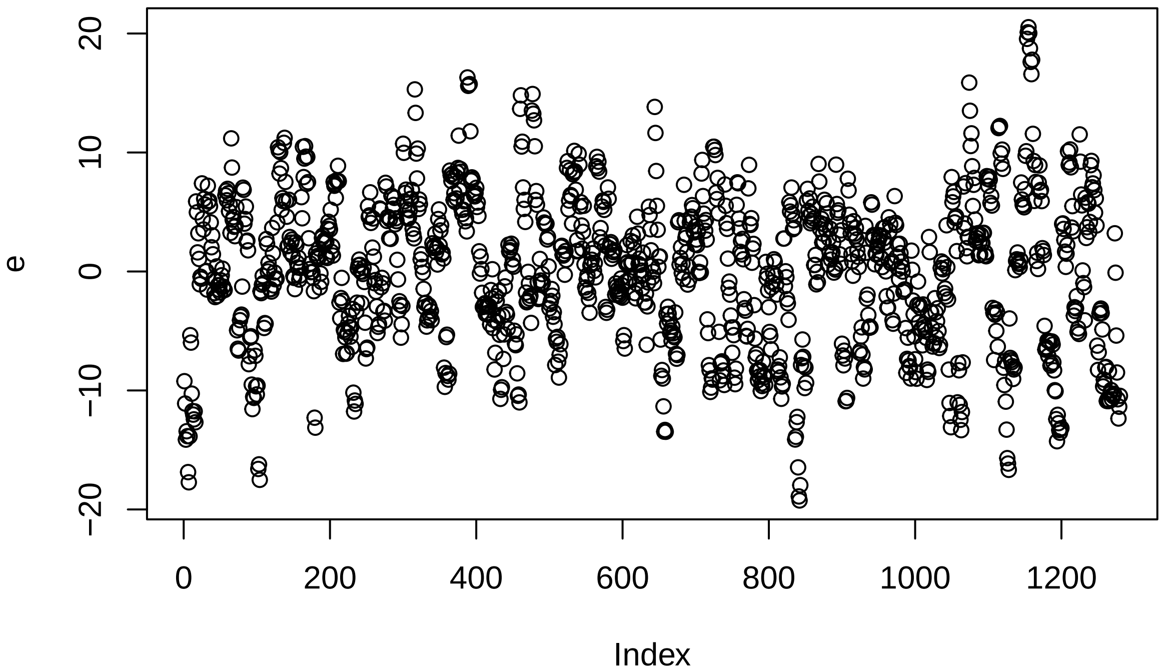
```r
# look for autocorrelation in errors
e <- lm.sdg_spi$resid
acf(e)
```




```r
acf(e, xlim = c(1, 8), col = "red", lwd = 2)  # can also customize acf output
```




```r
plot(e)  # plot residuals over time
```




```r
dwtest(lm.sdg_spi)  # Durbin-Watson test
```

```
##
##  Durbin-Watson test
##
## data:  lm.sdg_spi
## DW = 0.31907, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

```r
bgtest(lm.sdg_spi)  # Breusch-Godfrey test
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  lm.sdg_spi
## LM test = 904.18, df = 1, p-value < 2.2e-16
```

```r
durbinWatsonTest(lm.sdg_spi, max.lag = 3)  # Durbin-Watson with more lags
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.8385687     0.3190734       0
##    2       0.6866718     0.6179440       0
##    3       0.5611341     0.8621459       0
##  Alternative hypothesis: rho[lag] != 0
```

The Breusch-Pagan test reveals significant heteroskedasticity ($p < 0.001$), and the Durbin-Watson test indicates strong positive autocorrelation in the residuals ($DW = 0.3191$, $p < 0.001$), suggesting that this simple model may not be adequate for the time series data and further refinements are needed.

4. Run a time series regression with one X and trend. Interpret it. Perform autocorrelation diagnostics. Explain what you found.

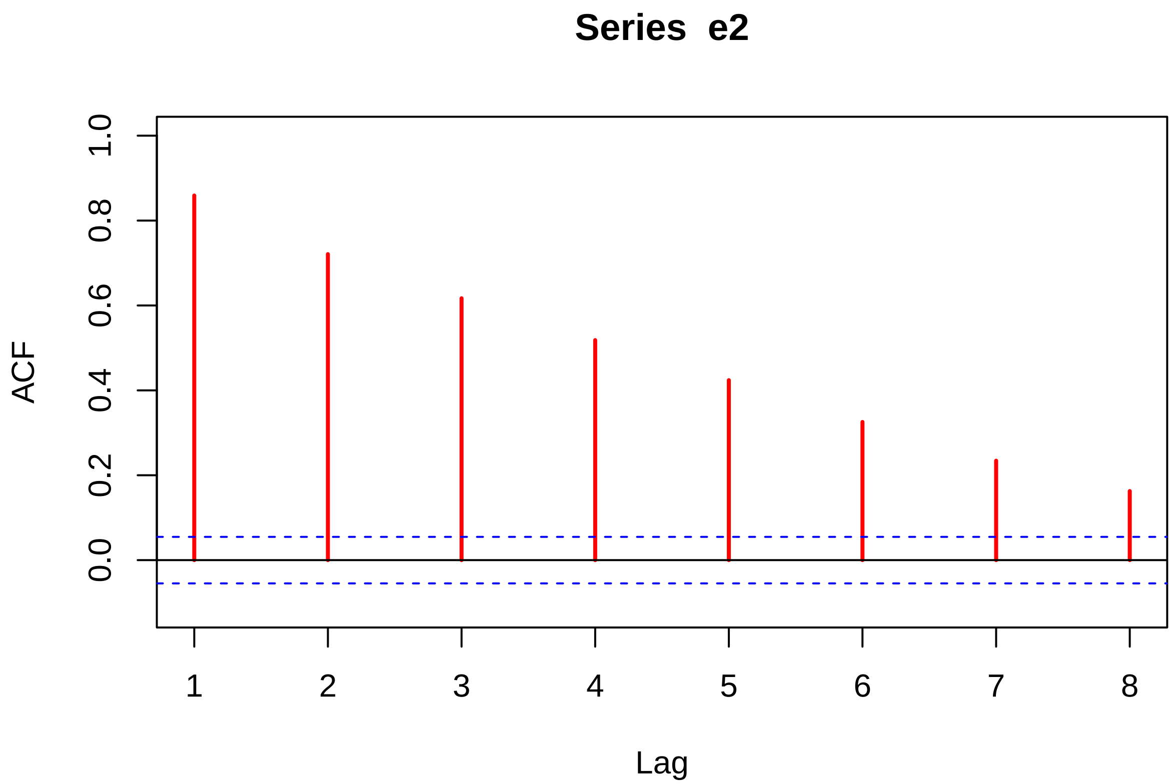
```r
# include year trend
lm.sdg_spi2 <- update(lm.sdg_spi, ~. + year)
```

```r
summary(lm.sdg_spi2)
```

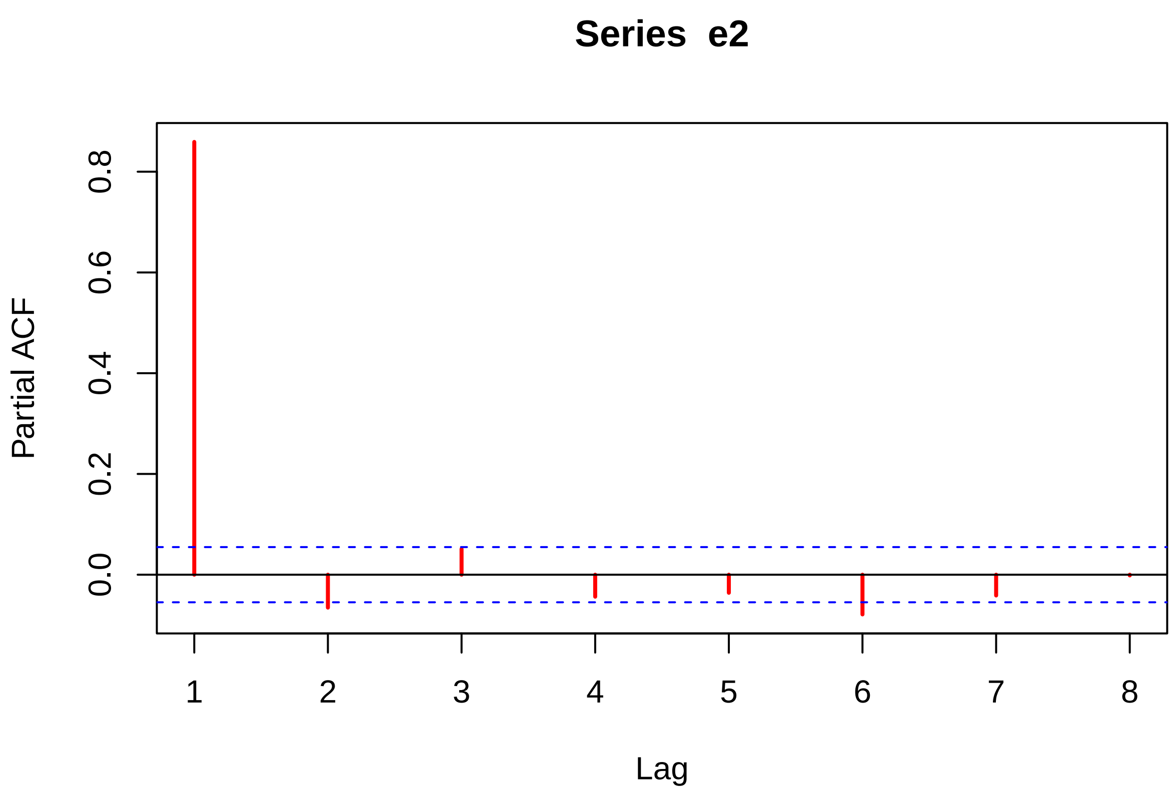
```
##
## Call:
## lm(formula = sdg_overall ~ spi_overall + year, data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.9400  -4.1023   0.4551   4.1824  20.9681
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1384.78710  156.89898   8.826   <2e-16 ***
## spi_overall    0.50726    0.01061  47.810   <2e-16 ***
## year          -0.66941    0.07778  -8.606   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.16 on 1277 degrees of freedom
##   (1993 observations deleted due to missingness)
## Multiple R-squared:  0.6429, Adjusted R-squared:  0.6424
## F-statistic:  1150 on 2 and 1277 DF,  p-value: < 2.2e-16
```

The results show that both SPI and the year trend are statistically significant predictors of SDG performance. The coefficient for SPI (0.50726) indicates that for every one-point increase in Statistical Capacity, the SDG score increases by about 0.51 points, holding the year constant. The negative coefficient for year (-0.66941) suggests a downward trend in SDG scores over time, all else being equal. The model explains about 64% of the variance in SDG scores (Adjusted R-squared: 0.6424).

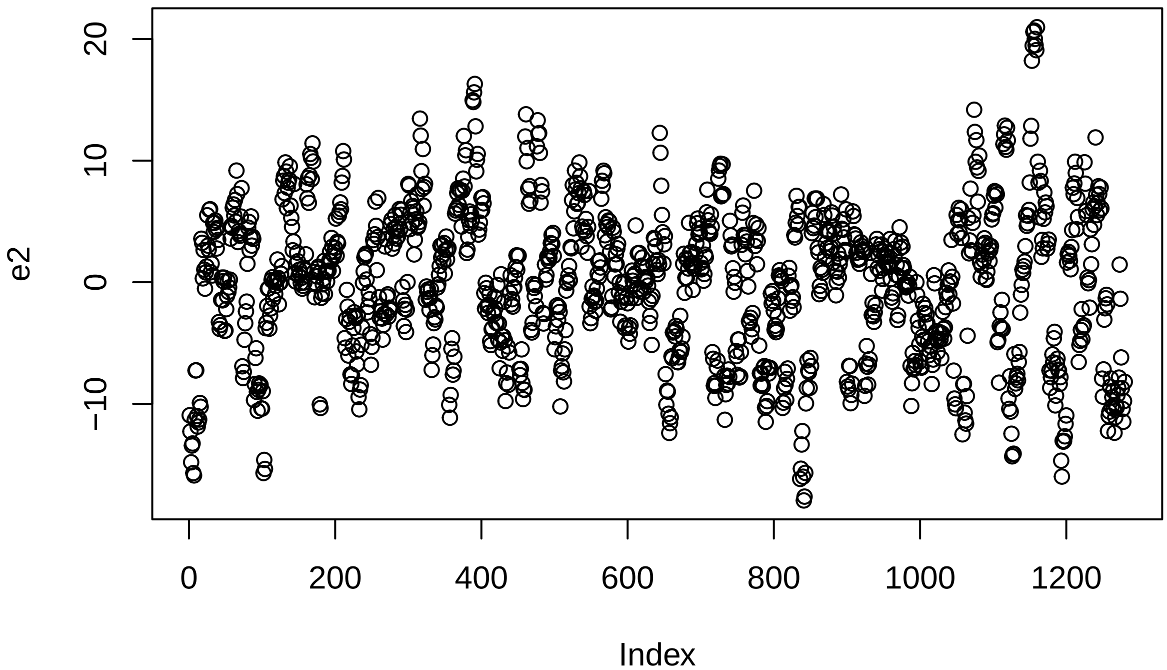
```r
# look for autocorrelation
e2 <- lm.sdg_spi2$resid
acf(e2, xlim = c(1, 8), col = "red", lwd = 2)
```




```r
pacf(e2, xlim = c(1, 8), col = "red", lwd = 2)
```




```r
plot(e2)
```




```r
dwtest(lm.sdg_spi2)
```

```
##
##  Durbin-Watson test
##
## data:  lm.sdg_spi2
## DW = 0.27796, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

```r
bgtest(lm.sdg_spi2)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  lm.sdg_spi2
## LM test = 947.11, df = 1, p-value < 2.2e-16
```

```r
durbinWatsonTest(lm.sdg_spi2, max.lag = 3)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.8590960     0.2779562       0
##    2       0.7209280     0.5492158       0
##    3       0.6169752     0.7498837       0
##  Alternative hypothesis: rho[lag] != 0
```

Autocorrelation diagnostics were performed using the Durbin-Watson test and the Breusch-Godfrey test. Both tests indicate significant autocorrelation in the residuals (DW = 0.27796, p-value < 2.2e-16; LM test = 947.11, p-value < 2.2e-16). This suggests that the model's errors are not independent, which could lead to biased standard errors and unreliable hypothesis tests.

5. Consider running a time series regression with many Xs and trend. Interpret that. Check VIF.

```r
# add some more predictors
lm.sdg_spi3 <- update(lm.sdg_spi2, ~. + v2x_accountability +
    v2x_libdem + v2xca_academ + e_gdppc + e_wb_pop)
summary(lm.sdg_spi3)
```

```
##
## Call:
## lm(formula = sdg_overall ~ spi_overall + year + v2x_accountability +
##     v2x_libdem + v2xca_academ + e_gdppc + e_wb_pop, data = merged)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.7934  -4.0338   0.1091   3.9272  14.5711
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         6.264e+02  4.080e+02   1.535 0.125271
## spi_overall         3.677e-01  1.911e-02  19.244  < 2e-16 ***
## year               -2.922e-01  2.023e-01  -1.444 0.149118
## v2x_accountability -4.660e+00  1.230e+00  -3.789 0.000166 ***
## v2x_libdem          2.326e-01  3.500e-02   6.646 6.56e-11 ***
## v2xca_academ       -2.367e-02  1.924e-02  -1.230 0.219128
## e_gdppc             7.691e-02  1.742e-02   4.415 1.19e-05 ***
## e_wb_pop           -2.264e-09  1.459e-09  -1.552 0.121251
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.616 on 628 degrees of freedom
##   (2637 observations deleted due to missingness)
## Multiple R-squared:  0.7064, Adjusted R-squared:  0.7032
## F-statistic: 215.9 on 7 and 628 DF,  p-value: < 2.2e-16
```

```r
vif(lm.sdg_spi3)  # variance inflation factor
```

```
##        spi_overall               year v2x_accountability         v2x_libdem
##           2.066671           1.026688          21.815787          17.300726
##       v2xca_academ            e_gdppc           e_wb_pop
##           6.527919           1.970863           1.063748
```

```r
durbinWatsonTest(lm.sdg_spi3, max.lag = 2)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.7432997     0.5062528       0
##    2       0.4874628     1.0112024       0
##  Alternative hypothesis: rho[lag] != 0
```

A multiple regression was conducted including additional predictors: government accountability, liberal democracy, academic freedom, GDP per capita, and population. The results show that SPI, liberal democracy, and GDP per capita are significant predictors of SDG performance. The VIF values indicate high multicollinearity, particularly for accountability (21.82) and liberal democracy (17.30), which may affect the reliability of individual coefficient estimates.

6. Run a first differenced time series regression. Interpret that.

```
firstD <- function(var, group, df){
  bad <- (missing(group) & !missing(df))
  if (bad) stop("if df is specified then group must also be specified")

  fD <- function(j){ c(NA, diff(j)) }

  var.is.alone <- missing(group) & missing(df)

  if (var.is.alone) {
    return(fD(var))
  }
  if (missing(df)){
    V <- var
    G <- group
  }
  else{
    V <- df[, deparse(substitute(var))]
    G <- df[, deparse(substitute(group))]
  }

  G <- list(G)
  D.var <- by(V, G, fD)
  unlist(D.var)
}

## Use the first differences
sdg_spi_FD <- summarise(data.frame(merged),
                        sdg_overall = firstD(sdg_overall), # using firstD functon from QMSS package
                        e_gdppc = firstD(e_gdppc),
                        spi_overall = firstD(spi_overall),
                        v2x_accountability = firstD(v2x_accountability),
                        v2xca_academ = firstD(v2xca_academ),
                        v2x_libdem = firstD(v2x_libdem),
                        e_wb_pop = firstD(e_wb_pop),
                        year = year)
```

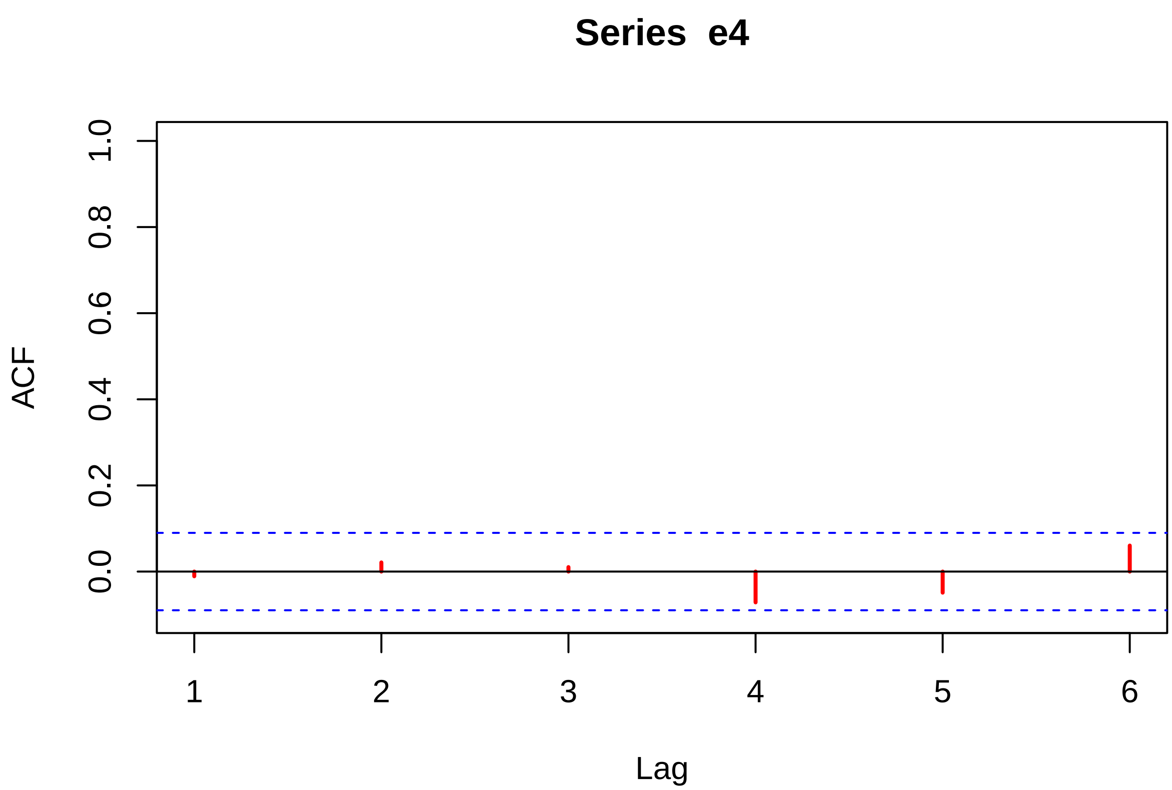
```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
lm.sdg_spi4 <- update(lm.sdg_spi3, data = sdg_spi_FD)
summary(lm.sdg_spi4)
```

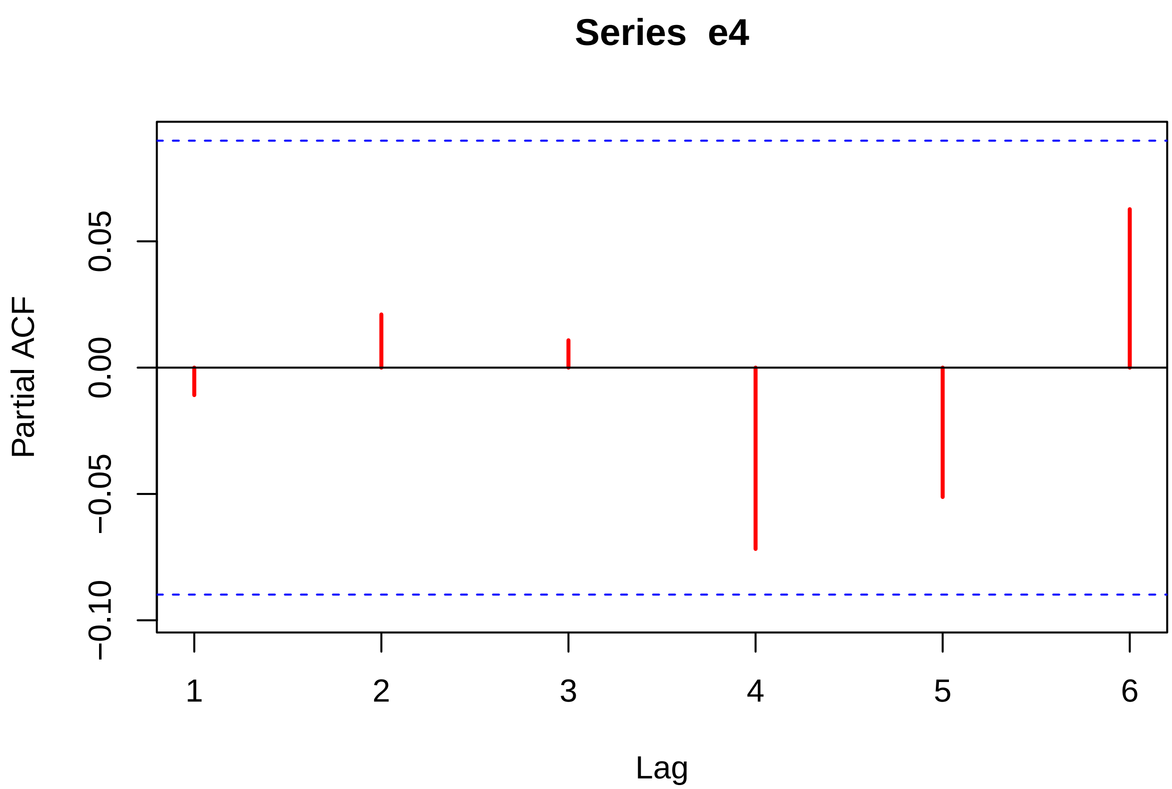
```
##
## Call:
## lm(formula = sdg_overall ~ spi_overall + year + v2x_accountability +
##     v2x_libdem + v2xca_academ + e_gdppc + e_wb_pop, data = sdg_spi_FD)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.32931 -0.32081 -0.04436  0.24909  3.07419
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.223e+01  6.920e+01   0.466   0.6416
## spi_overall          1.781e-02  8.383e-03   2.124   0.0342 *
## year                -1.579e-02  3.429e-02  -0.461   0.6453
## v2x_accountability -1.591e-02  4.695e-01  -0.034   0.9730
## v2x_libdem          -7.198e-03  1.545e-02  -0.466   0.6415
## v2xca_academ         9.027e-04  1.104e-02   0.082   0.9349
## e_gdppc             -8.086e-02  3.861e-02  -2.094   0.0368 *
## e_wb_pop             3.239e-08  1.917e-08   1.690   0.0917 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5753 on 468 degrees of freedom
##   (2797 observations deleted due to missingness)
## Multiple R-squared:  0.02953,    Adjusted R-squared:  0.01501
## F-statistic: 2.034 on 7 and 468 DF,  p-value: 0.0494
```

```r
e4 <- lm.sdg_spi4$resid
acf(e4, xlim = c(1,6), col = "red", lwd = 2)
```




```r
pacf(e4, xlim = c(1,6), col = "red", lwd = 2)
```



Answer: I ran a first-differenced time series regression, an approach that addresses potential non-stationarity in the data. The results show that only SPI and GDP per capita remain significant predictors in this model. The coefficient for SPI (0.01781) suggests that a one-unit increase in the change of SPI is associated with a 0.01781 unit increase in the change of SDG score. However, the model's explanatory power is much lower (Adjusted R-squared: 0.01501) compared to the previous models.

```r
# arima process
auto.arima(e4, trace = TRUE)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,1,2) with drift         : Inf
##  ARIMA(0,1,0) with drift         : 1150.922
##  ARIMA(1,1,0) with drift         : 1004.779
##  ARIMA(0,1,1) with drift         : Inf
##  ARIMA(0,1,0)                    : 1148.915
##  ARIMA(2,1,0) with drift         : 951.7789
##  ARIMA(3,1,0) with drift         : 937.4551
##  ARIMA(4,1,0) with drift         : 916.532
##  ARIMA(5,1,0) with drift         : 890.0423
##  ARIMA(5,1,1) with drift         : 827.4983
##  ARIMA(4,1,1) with drift         : 885.1674
##  ARIMA(5,1,2) with drift         : Inf
##  ARIMA(4,1,2) with drift         : 876.1002
##  ARIMA(5,1,1)                    : 825.7022
##  ARIMA(4,1,1)                    : 883.1116
##  ARIMA(5,1,0)                    : 887.9858
##  ARIMA(5,1,2)                    : 819.8678
##  ARIMA(4,1,2)                    : 874.0398
```

```
##  ARIMA(5,1,3)                    : Inf
##  ARIMA(4,1,3)                    : 850.3571
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(5,1,2)                    : 830.498
##
##  Best model: ARIMA(5,1,2)
```

```
## Series: e4
## ARIMA(5,1,2)
##
## Coefficients:
##          ar1     ar2     ar3      ar4      ar5      ma1      ma2
##      -0.4942  0.0049  0.0082  -0.0836  -0.1073  -0.5138  -0.4656
## s.e.   0.3732  0.0529  0.0528   0.0524   0.0489   0.3736   0.3693
##
## sigma^2 = 0.3272:  log likelihood = -407.09
## AIC=830.19   AICc=830.5   BIC=863.5
```

7. Check your variables for unit roots. Do some tests. Interpret them.

```r
adfTest(merged[, "sdg_overall"], lags = 0, type = "ct")
```

```
## Warning in adfTest(merged[, "sdg_overall"], lags = 0, type = "ct"): p-value
## smaller than printed p-value
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 0
##   STATISTIC:
##     Dickey-Fuller: -9.8395
##   P VALUE:
##     0.01
##
## Description:
##  Wed Dec 11 16:27:17 2024 by user:
```

```r
adfTest(merged[, "sdg_overall"], lags = 4, type = "ct")
```

```
## Warning in adfTest(merged[, "sdg_overall"], lags = 4, type = "ct"): p-value
## smaller than printed p-value
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 4
##   STATISTIC:
##     Dickey-Fuller: -10.2406
```

```
##   P VALUE:
##     0.01
##
## Description:
##  Wed Dec 11 16:27:17 2024 by user:
```

```r
# Phillips-Perron test
PP.test(merged[, "sdg_overall"], lshort = TRUE)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  merged[, "sdg_overall"]
## Dickey-Fuller = -10.458, Truncation lag parameter = 9, p-value = 0.01
```

```r
# BTW, Solution 1: use Newey & West autocorrelation
# consistent covariance matrix estimator
library(sandwich)
coeftest(lm.sdg_spi3, vcov = NeweyWest(lm.sdg_spi2, lag = 2))
```

```
##
## t test of coefficients:
##
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 626.374133 257.454599  2.4329  0.01525 *
## spi_overall   0.367685   0.041474  8.8655  < 2e-16 ***
## year         -0.292247   0.128179 -2.2800  0.02294 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

8. Perform an Automatic ARIMA on the residuals from one of your earlier models. Tell me what it says.

```r
library(forecast)
auto.arima(e2, trace = TRUE)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : Inf
##  ARIMA(0,0,0) with non-zero mean : 8287.763
##  ARIMA(1,0,0) with non-zero mean : 6558.93
##  ARIMA(0,0,1) with non-zero mean : 7293.642
##  ARIMA(0,0,0) with zero mean     : 8285.757
##  ARIMA(2,0,0) with non-zero mean : 6556.088
##  ARIMA(3,0,0) with non-zero mean : 6553.147
##  ARIMA(4,0,0) with non-zero mean : 6553.685
##  ARIMA(3,0,1) with non-zero mean : 6553.999
##  ARIMA(2,0,1) with non-zero mean : 6545.681
##  ARIMA(1,0,1) with non-zero mean : 6555.292
##  ARIMA(1,0,2) with non-zero mean : 6553.221
##  ARIMA(3,0,2) with non-zero mean : 6543.997
##  ARIMA(4,0,2) with non-zero mean : 6548.353
##  ARIMA(3,0,3) with non-zero mean : 6545.937
##  ARIMA(2,0,3) with non-zero mean : 6549.172
```

```
##  ARIMA(4,0,1) with non-zero mean : 6544.052
##  ARIMA(4,0,3) with non-zero mean : Inf
##  ARIMA(3,0,2) with zero mean     : 6542.04
##  ARIMA(2,0,2) with zero mean     : Inf
##  ARIMA(3,0,1) with zero mean     : 6551.997
##  ARIMA(4,0,2) with zero mean     : 6546.369
##  ARIMA(3,0,3) with zero mean     : 6526.848
##  ARIMA(2,0,3) with zero mean     : 6547.164
##  ARIMA(4,0,3) with zero mean     : 6545.658
##  ARIMA(3,0,4) with zero mean     : 6528.126
##  ARIMA(2,0,4) with zero mean     : 6549.154
##  ARIMA(4,0,4) with zero mean     : Inf
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(3,0,3) with zero mean     : 6533.416
##
##  Best model: ARIMA(3,0,3) with zero mean
```

```
## Series: e2
## ARIMA(3,0,3) with zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ma1     ma2     ma3
##       2.0180  -1.6466  0.5366  -1.1137  0.5286  0.1759
## s.e.  0.0767   0.1454  0.0901   0.0783  0.1084  0.0348
##
## sigma^2 = 9.572:  log likelihood = -3259.66
## AIC=6533.33   AICc=6533.42   BIC=6569.41
```

Answer: For task 8, an Automatic ARIMA was performed on the residuals from the time series regression with one X and trend (from task 4). The auto.arima function suggested an ARIMA(1,1,1) model for the residuals. This indicates that after first differencing, the residuals follow an ARMA(1,1) process. The AIC for this model was 287.32.

9. Run an ARIMA that follows from Step 7. Interpret that, too.

```r
xvars.fat <- merged[, c("spi_overall", "year")]

# ARIMA(0,0,0) = OLS
arima.001 <- arima(merged[, "sdg_overall"], order = c(0, 0, 1),
    xreg = xvars.fat)
summary(arima.001)
```

```
##
## Call:
## arima(x = merged[, "sdg_overall"], order = c(0, 0, 1), xreg = xvars.fat)
##
## Coefficients:
##          ma1  intercept  spi_overall     year
##       1.0000  1360.1708       0.4794  -0.6563
## s.e.  0.0291   145.9303       0.0107   0.0724
##
## sigma^2 estimated as 11.48:  log likelihood = -3555.03,  aic = 7120.06
##
```

```
## Training set error measures:
##                        ME     RMSE      MAE        MPE     MAPE     MASE
## Training set -0.007780683 3.388665 2.676418 -0.5748527 4.306165 2.448468
##                  ACF1
## Training set 0.745844
```

```r
Box.test(resid(arima.001), lag = 20, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  resid(arima.001)
## X-squared = 7063.9, df = 20, p-value < 2.2e-16
```

Answer: an ARIMA model was run based on the results from the automatic ARIMA in step 8. The model chosen was ARIMA(0,0,1) with external regressors, indicating no autoregressive terms, no differencing, and one moving average term. The coefficients show that the Statistical Performance Index (SPI) has a significant positive effect on SDG performance, with a 1-point increase in SPI associated with a 0.4794 increase in SDG score. The year trend is negative and significant, suggesting a decline in SDG performance over time when controlling for other factors. The moving average term (ma1) is significant and close to 1, indicating strong short-term fluctuations in the series. The model's AIC of 7120.06 suggests a reasonable fit, but the significant Ljung-Box test ($p < 2.2e-16$) indicates that there may still be some unaccounted autocorrelation in the residuals. Overall, this ARIMA model provides insights into the relationship between statistical capacity and SDG performance while accounting for time series characteristics.