ENV 790.30 - Time Series Analysis for Energy Data | Spring 2025

Assignment 3 - Due date 02/04/25

Rosie Wu

Directions

You should open the .rmd file corresponding to this assignment on RStudio. The file is available on our class repository on Github.

Once you have the file open on your local machine the first thing you will do is rename the file such that it includes your first and last name (e.g., "LuanaLima_TSA_A03_Sp25.Rmd"). Then change "Student Name" on line 4 with your name.

Then you will start working through the assignment by **creating code and output** that answer each question. Be sure to use this assignment document. Your report should contain the answer to each question and any plots/tables you obtained (when applicable).

Please keep this R code chunk options for the report. It is easier for us to grade when we can see code and output together. And the tidy.opts will make sure that line breaks on your code chunks are automatically added for better visualization.

When you have completed the assignment, **Knit** the text and code into a single PDF file. Submit this pdf using Sakai.

Questions

Consider the same data you used for A2 from the spreadsheet "Table_10.1_Renewable_Energy_Production_and_Consumpti
The data comes from the US Energy Information and Administration and corresponds to the December 2024 **Monthly** Energy Review. Once again you will work only with the following columns: Total Renewable Energy Production and Hydroelectric Power Consumption. Create a data frame structure with these two time series only.

R packages needed for this assignment:"forecast","tseries", and "Kendall". Install these packages, if you haven't done yet. Do not forget to load them before running your script, since they are NOT default packages.\

```
#Load/install required package here
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```r
library(ggplot2)
library(forecast)
```

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

```r
library(Kendall)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

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

```r
library(here)
```

```
## here() starts at /home/guest/TSA_Sp25
```

```r
library(readxl)
library(openxlsx)
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```r
#library(tseries)
#  Console message: namespace 'tseries' is imported by 'forecast'
# so cannot be unloaded

#dataset setup
#Importing data set
here()
```

```
## [1] "/home/guest/TSA_Sp25"
```

```r
#Importing data set without change the original file using read.xlsx
energy_data1 <- read_excel(path= "./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Sour
                           skip = 12, sheet="Monthly Data",col_names=FALSE)
```

```
## New names:
## * `` -> `...1`
## * `` -> `...2`
## * `` -> `...3`
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...6`
## * `` -> `...7`
## * `` -> `...8`
## * `` -> `...9`
## * `` -> `...10`
## * `` -> `...11`
## * `` -> `...12`
## * `` -> `...13`
## * `` -> `...14`
```

```
#Now let's extract the column names from row 11
read_col_names <- read_excel(
  path="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
  skip = 10,n_max = 1, sheet="Monthly Data",col_names=FALSE)
```

```
## New names:
## * `` -> `...1`
## * `` -> `...2`
## * `` -> `...3`
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...6`
## * `` -> `...7`
## * `` -> `...8`
## * `` -> `...9`
## * `` -> `...10`
## * `` -> `...11`
## * `` -> `...12`
## * `` -> `...13`
## * `` -> `...14`
```

```
#Assign the column names to the data set
colnames(energy_data1) <- read_col_names

#Visualize the first rows of the data set
head(energy_data1)
```

```
## # A tibble: 6 x 14
##   Month               `Wood Energy Production` `Biofuels Production`
##   <dttm>                                 <dbl> <chr>
## 1 1973-01-01 00:00:00                     130. Not Available
## 2 1973-02-01 00:00:00                     117. Not Available
## 3 1973-03-01 00:00:00                     130. Not Available
## 4 1973-04-01 00:00:00                     125. Not Available
## 5 1973-05-01 00:00:00                     130. Not Available
## 6 1973-06-01 00:00:00                     125. Not Available
## # i 11 more variables: `Total Biomass Energy Production` <dbl>,
```

```
## #   `Total Renewable Energy Production` <dbl>,
## #   `Hydroelectric Power Consumption` <dbl>,
## #   `Geothermal Energy Consumption` <dbl>, `Solar Energy Consumption` <chr>,
## #   `Wind Energy Consumption` <chr>, `Wood Energy Consumption` <dbl>,
## #   `Waste Energy Consumption` <dbl>, `Biofuels Consumption` <chr>,
## #   `Total Biomass Energy Consumption` <dbl>, ...
```

```r
# select columns
energy_df3_selected <- energy_data1 %>%
  select("Month",
         "Total Renewable Energy Production",
         "Hydroelectric Power Consumption")
energy_df3_selected$Month <- as.Date(energy_df3_selected$Month)
head(energy_df3_selected)
```

```
## # A tibble: 6 x 3
##   Month      `Total Renewable Energy Production` Hydroelectric Power Consumpti~1
##   <date>                                   <dbl>                            <dbl>
## 1 1973-01-01                                220.                             89.6
## 2 1973-02-01                                197.                             79.5
## 3 1973-03-01                                219.                             88.3
## 4 1973-04-01                                209.                             83.2
## 5 1973-05-01                                216.                             85.6
## 6 1973-06-01                                208.                             82.1
## # i abbreviated name: 1: `Hydroelectric Power Consumption`
```

##Trend Component

Q1

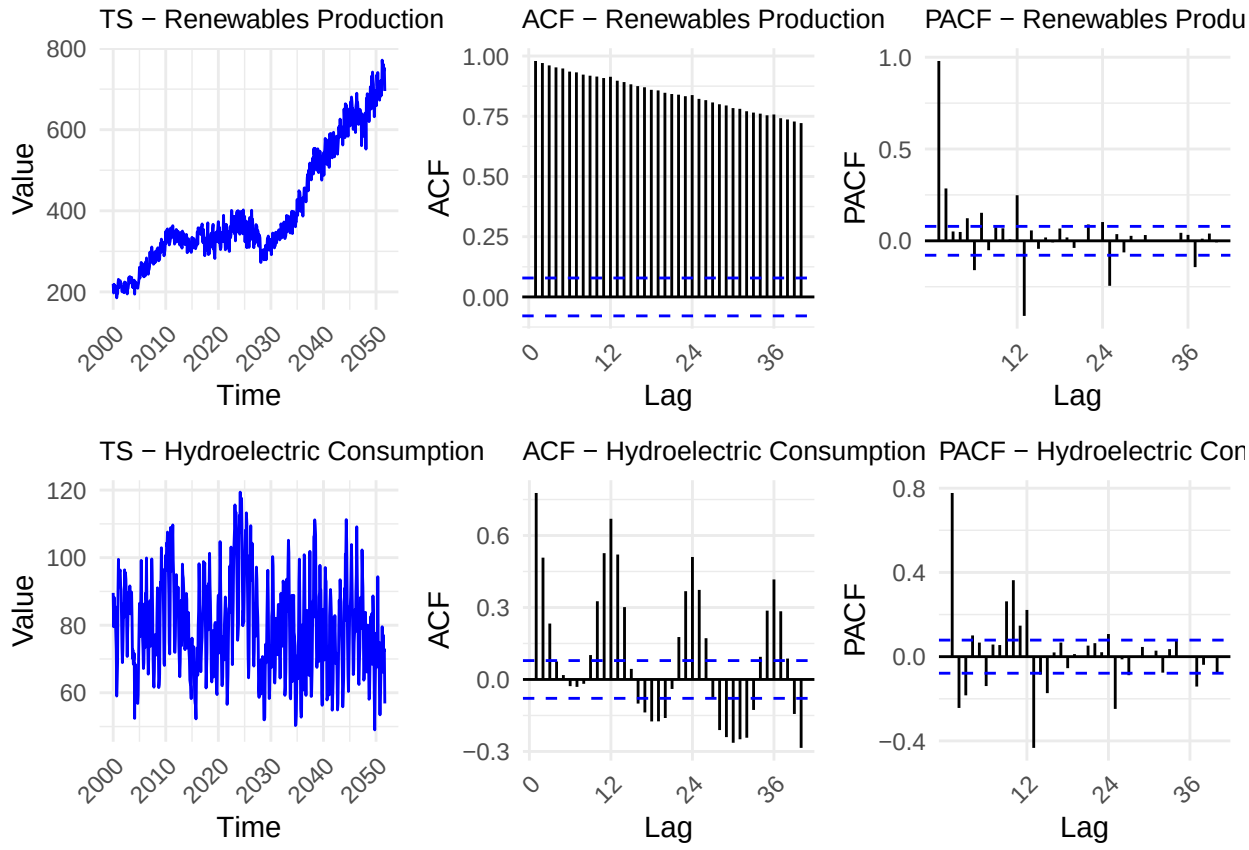
For each time series, i.e., Renewable Energy Production and Hydroelectric Consumption create three plots: one with time series, one with the ACF and with the PACF. You may use the some code form A2, but I want all the three plots side by side as in a grid. (Hint: use function `plot_grid()` from the `cowplot` package)

```r
renewable_ts <- ts(energy_df3_selected[,2],start=c(2000,1),frequency=12)
hydro_ts <- ts(energy_df3_selected[,3],start=c(2000,1),frequency=12)
```

I was having a hard time grasping the for loops ie. the for i in nobs or nhydros, so I tried a different approach by creating a large function for the - ts dataframes and plots, then return these plots together in a matrix form. In plot_grid, I will index the plots from this large function to extract the plot I need.

```r
# Function to create time series, ACF, and PACF plots
plot_time_series_acf_pacf <- function(ts_data, ts_name) {
  # Convert time series to data frame for ggplot
  ts_df <- data.frame(Date = time(ts_data), Value = as.numeric(ts_data))

  ts_plot <- ggplot(ts_df, aes(x = Date, y = Value)) +
    geom_line(color = "blue") +
    ggtitle(paste("TS -", ts_name)) +
    xlab("Time") + ylab("Value") +
    theme_minimal() +
    theme(
```

```
      plot.title = element_text(size = 10),
      axis.text.x = element_text(angle = 45, hjust = 1))
  # Rotate x-axis labels to fix

  acf_plot <- ggAcf(ts_data, lag.max = 40) +
    ggtitle(paste("ACF -", ts_name)) +
    theme_minimal() +
    theme(
      plot.title = element_text(size = 10),
      axis.text.x = element_text(angle = 45, hjust = 1))

  pacf_plot <- ggPacf(ts_data, lag.max = 40) +
    ggtitle(paste("PACF -", ts_name)) +
    theme_minimal() +
   theme(
      plot.title = element_text(size = 10),
      axis.text.x = element_text(angle = 45, hjust = 1))


  return(list(ts_plot, acf_plot, pacf_plot))}

# Generate plots for Renewable Energy Production and Hydroelectric Consumption
renewable_plots <- plot_time_series_acf_pacf(renewable_ts,
                                             "Renewables Production")
hydro_plots <- plot_time_series_acf_pacf(hydro_ts,
                                         "Hydroelectric Consumption")

# Combine all plots into a 2-row grid (one row per variable, each with 3 plots)
final_plot1 <- plot_grid(
  plot_grid(plotlist = renewable_plots, ncol = 3),
  plot_grid(plotlist = hydro_plots, ncol = 3),
  nrow = 2
)

# Display the final plot
print(final_plot1)
```



Q2

From the plot in Q1, do the series Total Renewable Energy Production and Hydroelectric Power Consumption appear to have a trend? If yes, what kind of trend?

> Answer: By observing the ts plots in Q1, the Renewable Energy Production seems to have a relatively consistent upward linear trend, whereas the Hydroelectric Consumption does not appear to have a visible trend overtime. According to the trend of the TS and ACF plots of Renewables Production, it shows relatively a non-stationary trend. As the lag increases, ACF value is also decreasing, this could mean the correlation/ trend may not last as long or stay consistent for long. There seems to be a very subtle seasonal spike in the ACF of Renewables in the first 24 lags. Overtime, the value decreased dramtically till almost 0. In the plots of Hydroelectric ACF and PACFs, the seasonality appears to be more dramatic. Yet, there still displays decreasing in values and more negative values over lags. This could mean the correlation is decreasing overtime, and the trend and seasonality could be less consistent as it's further into the future.

Q3

Use the *lm()* function to fit a linear trend to the two time series. Ask R to print the summary of the regression. Interpret the regression output, i.e., slope and intercept. Save the regression coefficients for further analysis.

```r
# Fit linear trend using lm()
# Create time index
time_index <- as.numeric(time(renewable_ts))

# Fit regression model
renewable_lm <- lm(as.numeric(renewable_ts) ~ time_index)
print(summary(renewable_lm))  # Print summary for interpretation
```

```
##
## Call:
## lm(formula = as.numeric(renewable_ts) ~ time_index)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -151.11  -37.84   13.53   41.76  149.42
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.720e+04  3.360e+02  -51.17   <2e-16 ***
## time_index   8.687e+00  1.659e-01   52.37   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61.75 on 619 degrees of freedom
## Multiple R-squared:  0.8159, Adjusted R-squared:  0.8156
## F-statistic:  2743 on 1 and 619 DF,  p-value: < 2.2e-16
```

```r
# Fit regression model for Hydroelectric Consumption
hydro_lm <- lm(as.numeric(hydro_ts) ~ time_index)
summary(hydro_lm)
```

```
##
## Call:
## lm(formula = as.numeric(hydro_ts) ~ time_index)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.995 -10.422  -0.720   9.161  39.624
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 346.41279   76.08047   4.553 6.36e-06 ***
## time_index   -0.13173    0.03755  -3.508 0.000485 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.98 on 619 degrees of freedom
## Multiple R-squared:  0.01949,    Adjusted R-squared:  0.01791
## F-statistic:  12.3 on 1 and 619 DF,  p-value: 0.0004848
```

```r
# Save regression coefficients
renewable_coef <- coef(renewable_lm)  # Intercept and slope for Renewable Energy
```

```r
hydro_coef <- coef(hydro_lm)  # Intercept and slope for Hydro Consumption

print(renewable_coef)
```

```
##   (Intercept)    time_index
## -17196.840061      8.687218
```

```r
print(hydro_coef)
```

```
## (Intercept)  time_index
## 346.4127948  -0.1317281
```

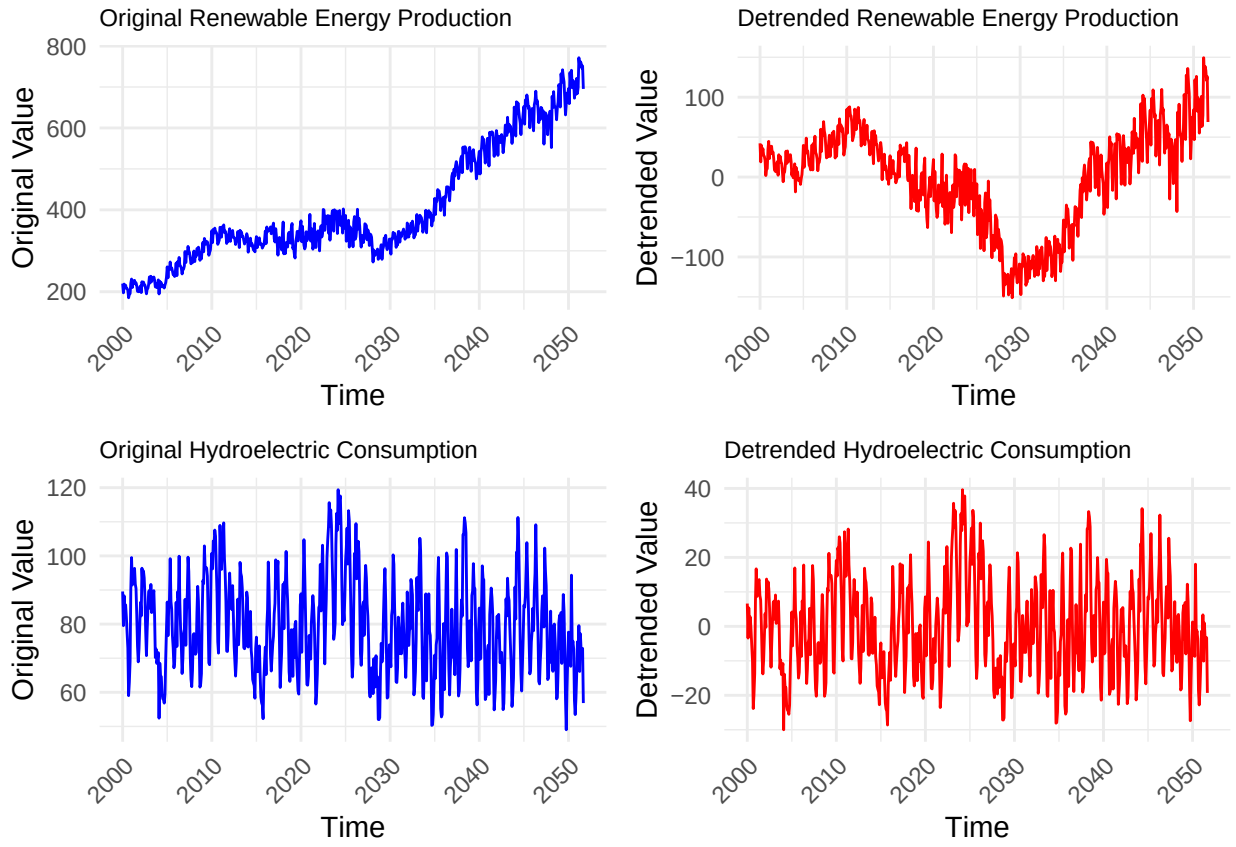
> Interpret slope and intercepts: Renewable Energy Production's slope for the regression is about 8.69, which means as the time series increases by each month by about 8.69. The intercept (when time_index = 0) is highly negative (-17196.84), which means it's the estimated value of the time series at the beginning of the dataset, and it shouldn't have real application here? For Hydro, the slope is much slower and is negative, which means there is not as much estimated change (trend) per unit time, if there is, it is slightly decreasing overtime. The intercept for hydro is a high value (346.41), which could mean that the esimtated value of the time series of Hydroelectric Consumption starts from 346.41.

Q4

Use the regression coefficients from Q3 to detrend the series. Plot the detrended series and compare with the plots from Q1. What happened? Did anything change?

```r
# Compute fitted trend values
renewable_trend <- renewable_coef[1] + renewable_coef[2] * time_index
hydro_trend <- hydro_coef[1] + hydro_coef[2] * time_index

# Detrend the series (orignal ts subtract trend)
renewable_detrended <- renewable_ts - renewable_trend
hydro_detrended <- hydro_ts - hydro_trend

# Convert to data frames for plotting
renewable_detrended_df <- data.frame(Date = time(renewable_ts), Value = as.numeric(renewable_detrended))
hydro_detrended_df <- data.frame(Date = time(hydro_ts), Value = as.numeric(hydro_detrended))

# Plot detrended series
renewable_detrended_plot <- ggplot(renewable_detrended_df, aes(x = Date, y = Value)) +
  geom_line(color = "red") +
  ggtitle("Detrended Renewable Energy Production") +
  xlab("Time") + ylab("Detrended Value") +
  theme_minimal() +
    theme(
      plot.title = element_text(size = 9),
      axis.text.x = element_text(angle = 45, hjust = 1))

hydro_detrended_plot <- ggplot(hydro_detrended_df, aes(x = Date, y = Value)) +
  geom_line(color = "red") +
  ggtitle("Detrended Hydroelectric Consumption") +
```

```
  xlab("Time") + ylab("Detrended Value") +
  theme_minimal()+
    theme(
      plot.title = element_text(size = 9),
      axis.text.x = element_text(angle = 45, hjust = 1))

# Compare with original time series
renewable_original_plot <- ggplot(data.frame(Date = time(renewable_ts), Value = as.numeric(renewable_ts)
                                  aes(x = Date, y = Value)) +
  geom_line(color = "blue") +
  ggtitle("Original Renewable Energy Production") +
  xlab("Time") + ylab("Original Value") +
  theme_minimal()+
    theme(
      plot.title = element_text(size = 9),
      axis.text.x = element_text(angle = 45, hjust = 1))


hydro_original_plot <- ggplot(data.frame(Date = time(hydro_ts), Value = as.numeric(hydro_ts)),
                              aes(x = Date, y = Value)) +
  geom_line(color = "blue") +
  ggtitle("Original Hydroelectric Consumption") +
  xlab("Time") + ylab("Original Value") +
  theme_minimal()+
    theme(
      plot.title = element_text(size = 9),
      axis.text.x = element_text(angle = 45, hjust = 1))


# Arrange all plots for comparison
final_comparison4 <- plot_grid(
  renewable_original_plot, renewable_detrended_plot,
  hydro_original_plot, hydro_detrended_plot,
  nrow = 2
)

# Display the final plot comparison
print(final_comparison4)
```



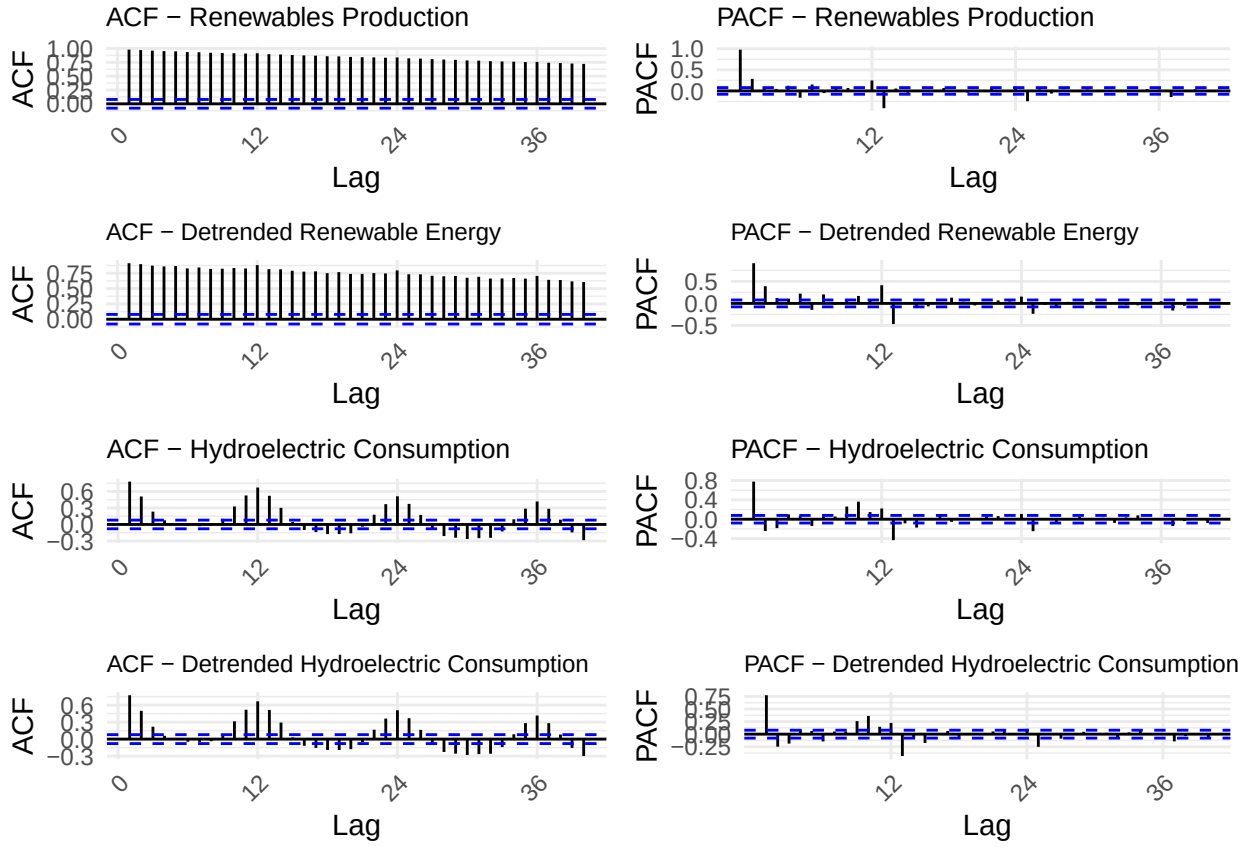
> Answer: The graphs for Hydroelectric Conumption did not change. In contrast, the detrended graph of Renewable Energy production showed a more visible change from the original model. It stil has a general increasing trend overtime, but there is a significant drop in 2030, adn the trend was slope down from year 2000 to 2030.

Q5

Plot ACF and PACF for the detrended series and compare with the plots from Q1. You may use plot_grid() again to get them side by side, but not mandatory. Did the plots change? How?

```
# Function to create ACF and PACF plots
plot_acf_pacf <- function(ts_data, ts_name) {
  acf_plot <- ggAcf(ts_data, lag.max = 40) +
    ggtitle(paste("ACF -", ts_name)) +
    theme_minimal()+
    theme(
      plot.title = element_text(size = 9),
      axis.text.x = element_text(angle = 45, hjust = 1))

  pacf_plot <- ggPacf(ts_data, lag.max = 40) +
    ggtitle(paste("PACF -", ts_name)) +
    theme_minimal()+
    theme(
      plot.title = element_text(size = 9),
      axis.text.x = element_text(angle = 45, hjust = 1))
```

```r
  return(list(acf_plot, pacf_plot))}

# Generate ACF and PACF plots for detrended series
renewable_detrended_plots <- plot_acf_pacf(renewable_detrended, "Detrended Renewable Energy")
hydro_detrended_plots <- plot_acf_pacf(hydro_detrended, "Detrended Hydroelectric Consumption")

# Compare original and detrended ACF/PACF plots using plot_grid
final_comparison <- plot_grid(
  renewable_plots[[2]], renewable_plots[[3]],
  renewable_detrended_plots[[1]], renewable_detrended_plots[[2]],
  hydro_plots[[2]], hydro_plots[[3]],
  hydro_detrended_plots[[1]], hydro_detrended_plots[[2]],
  nrow = 4)

# Display the final comparison plot
print(final_comparison)
```



> Answer: There was not much observable change between original and detrended ACF or PACF plots for Hydroelectric consumption. Since there was not much change before and after detrending (subtracting lm), it might indicate a lack of linear trend. There also did not observe much change in PACF for Renewable Production plots before and after detrending. There did display a slightly more seasonal pattern after detrending the ACF plot for Renewable Production.

Seasonal Component

Set aside the detrended series and consider the original series again from Q1 to answer Q6 to Q8.

Q6

Just by looking at the time series and the acf plots, do the series seem to have a seasonal trend? No need to run any code to answer your question. Just type in you answer below.

> Answer: According to the ACF graph of both renewables energy production and hydroelectric consumption, the detrended plots do show a slightly greater seasonal pattern. This contrast between original and detrended plots is more dramatic with Renewables Production, but seasonal pattern is generally more visible in Hydro plots. The detrended plots show a more consistent pattern after removing the trend, making the seasonal fluctuations more visible.

Q7

Use function *lm()* to fit a seasonal means model (i.e. using the seasonal dummies) the two time series. Ask R to print the summary of the regression. Interpret the regression output. From the results which series have a seasonal trend? Do the results match you answer to Q6?

```
# Extract frequency (e.g., 12 for monthly data), create time_index
freq <- frequency(renewable_ts, 12)
time_index <- as.numeric(time(renewable_ts))

# Create seasonal dummy variables
seasonal_dummies <- as.factor(cycle(renewable_ts))  # Extracts seasonality

# Fit seasonal means model for Renewable Energy Production
renewable_seasonal_lm <- lm(as.numeric(renewable_ts) ~ seasonal_dummies)
print(summary(renewable_seasonal_lm))
```

```
##
## Call:
## lm(formula = as.numeric(renewable_ts) ~ seasonal_dummies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -205.65  -91.59  -54.59  117.87  356.19
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        412.5713    20.0321  20.596   <2e-16 ***
## seasonal_dummies2  -36.3206    28.3296  -1.282    0.200
## seasonal_dummies3    2.7478    28.3296   0.097    0.923
## seasonal_dummies4   -9.8692    28.3296  -0.348    0.728
## seasonal_dummies5    5.2265    28.3296   0.184    0.854
## seasonal_dummies6   -5.3669    28.3296  -0.189    0.850
## seasonal_dummies7    0.8773    28.3296   0.031    0.975
## seasonal_dummies8   -6.4030    28.3296  -0.226    0.821
## seasonal_dummies9  -31.0100    28.3296  -1.095    0.274
## seasonal_dummies10 -22.1774    28.4681  -0.779    0.436
## seasonal_dummies11 -22.6784    28.4681  -0.797    0.426
## seasonal_dummies12  -1.9728    28.4681  -0.069    0.945
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 144.5 on 609 degrees of freedom
## Multiple R-squared:  0.008696,   Adjusted R-squared:  -0.009209
## F-statistic: 0.4857 on 11 and 609 DF,  p-value: 0.9126
```

```
# Fit seasonal means model for Hydroelectric Consumption
hydro_seasonal_lm <- lm(as.numeric(hydro_ts) ~ seasonal_dummies)
summary(hydro_seasonal_lm)
```

```
##
## Call:
## lm(formula = as.numeric(hydro_ts) ~ seasonal_dummies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.101  -6.241  -0.444   6.410  32.363
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         84.9218     1.4377  59.068  < 2e-16 ***
## seasonal_dummies2   -7.4540     2.0332  -3.666 0.000268 ***
## seasonal_dummies3    2.1119     2.0332   1.039 0.299346
## seasonal_dummies4    0.4578     2.0332   0.225 0.821940
## seasonal_dummies5    8.9655     2.0332   4.410 1.22e-05 ***
## seasonal_dummies6    5.7881     2.0332   2.847 0.004565 **
## seasonal_dummies7   -0.9083     2.0332  -0.447 0.655243
## seasonal_dummies8  -10.3406     2.0332  -5.086 4.88e-07 ***
## seasonal_dummies9  -21.5376     2.0332 -10.593  < 2e-16 ***
## seasonal_dummies10 -21.3107     2.0431 -10.430  < 2e-16 ***
## seasonal_dummies11 -15.7465     2.0431  -7.707 5.26e-14 ***
## seasonal_dummies12  -4.9412     2.0431  -2.418 0.015880 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.37 on 609 degrees of freedom
## Multiple R-squared:  0.4695, Adjusted R-squared:  0.4599
## F-statistic:    49 on 11 and 609 DF,  p-value: < 2.2e-16
```

```
# Extract regression coefficients
renewable_seasonal_coef <- coef(renewable_seasonal_lm)
hydro_seasonal_coef <- coef(hydro_seasonal_lm)

# Print coefficients
print(renewable_seasonal_coef)
```

```
##        (Intercept)  seasonal_dummies2  seasonal_dummies3  seasonal_dummies4
##        412.5712500        -36.3205769          2.7478269         -9.8691731
##  seasonal_dummies5  seasonal_dummies6  seasonal_dummies7  seasonal_dummies8
##          5.2264615         -5.3669038          0.8773462         -6.4029615
##  seasonal_dummies9 seasonal_dummies10 seasonal_dummies11 seasonal_dummies12
##        -31.0100192        -22.1773873        -22.6783873         -1.9727990
```

```
print(hydro_seasonal_coef)
```

```
##        (Intercept)  seasonal_dummies2  seasonal_dummies3  seasonal_dummies4
##         84.9218462         -7.4540000          2.1119231          0.4577692
##  seasonal_dummies5  seasonal_dummies6  seasonal_dummies7  seasonal_dummies8
##          8.9655000          5.7881346         -0.9082500        -10.3406346
##  seasonal_dummies9 seasonal_dummies10 seasonal_dummies11 seasonal_dummies12
##        -21.5376154        -21.3107089        -15.7464736         -4.9411991
```

Answer: If most coefficients are significant, the series has a strong seasonal pattern. According to the f-stats and p-value of the seasonal trend testing for renewables production and hydro consumption, the renewable energy production f-stats is too low (<1) and p-value (0.9) is too high than 95% or 99% confidence level. In contrast, the hydropower consumption seasonal trend coefficients testing f-stats is 49 and p-value is very close to 0. Therefore, Hydro shows a visible trend, but not renewables production. The more dramatic seasonal pattern for hydro is also reflected in Q6.

Q8

Use the regression coefficients from Q7 to deseason the series. Plot the deseason series and compare with the plots from part Q1. Did anything change?

```
# Function to deseasonalize a time series using stored coefficients
deseasonalize_ts_using_coeffs <- function(ts_data, seasonal_coeffs, start_year) {
  # Generate seasonal dummies
  dummies <- seasonaldummy(ts_data)

  # Compute seasonal component using stored coefficients
  seas_comp <- dummies %*% seasonal_coeffs[-1]  # Exclude intercept

  # Deseasonalize the data
  deseason_data <- as.numeric(ts_data) - seas_comp

  # Convert to time series object
  ts_deseason_data <- ts(deseason_data, start = c(start_year, 1), frequency = frequency(ts_data))

  return(ts_deseason_data)
}

# Get the starting year of the time series
start_year_renewable <- start(renewable_ts)[1]
start_year_hydro <- start(hydro_ts)[1]

# deseasonalization using stored coefficients
ts_deseason_renewable <- deseasonalize_ts_using_coeffs(
  renewable_ts, renewable_seasonal_coef, start_year_renewable)
ts_deseason_hydro <- deseasonalize_ts_using_coeffs(
  hydro_ts, hydro_seasonal_coef, start_year_hydro)

# Create data frames for plotting
renewable_deseasoned_df <- data.frame(Date = time(renewable_ts),
                                      Deseasonalized = as.numeric(ts_deseason_renewable))
```

```r
hydro_deseasoned_df <- data.frame(Date = time(hydro_ts),
                                  Deseasonalized = as.numeric(ts_deseason_hydro))
```

```r
# Create plots for Renewable Energy
renewable_deseasoned_plot <- ggplot(renewable_deseasoned_df, aes(x = Date)) +
  geom_line(aes(y = Deseasonalized), size = 1) +
  ggtitle("Deseasonalized Renewable Energy Production") +
  xlab("Time") + ylab("Energy Production") +
  theme_minimal()+
    theme(
      plot.title = element_text(size = 9),
      axis.text.x = element_text(angle = 45, hjust = 1))
```

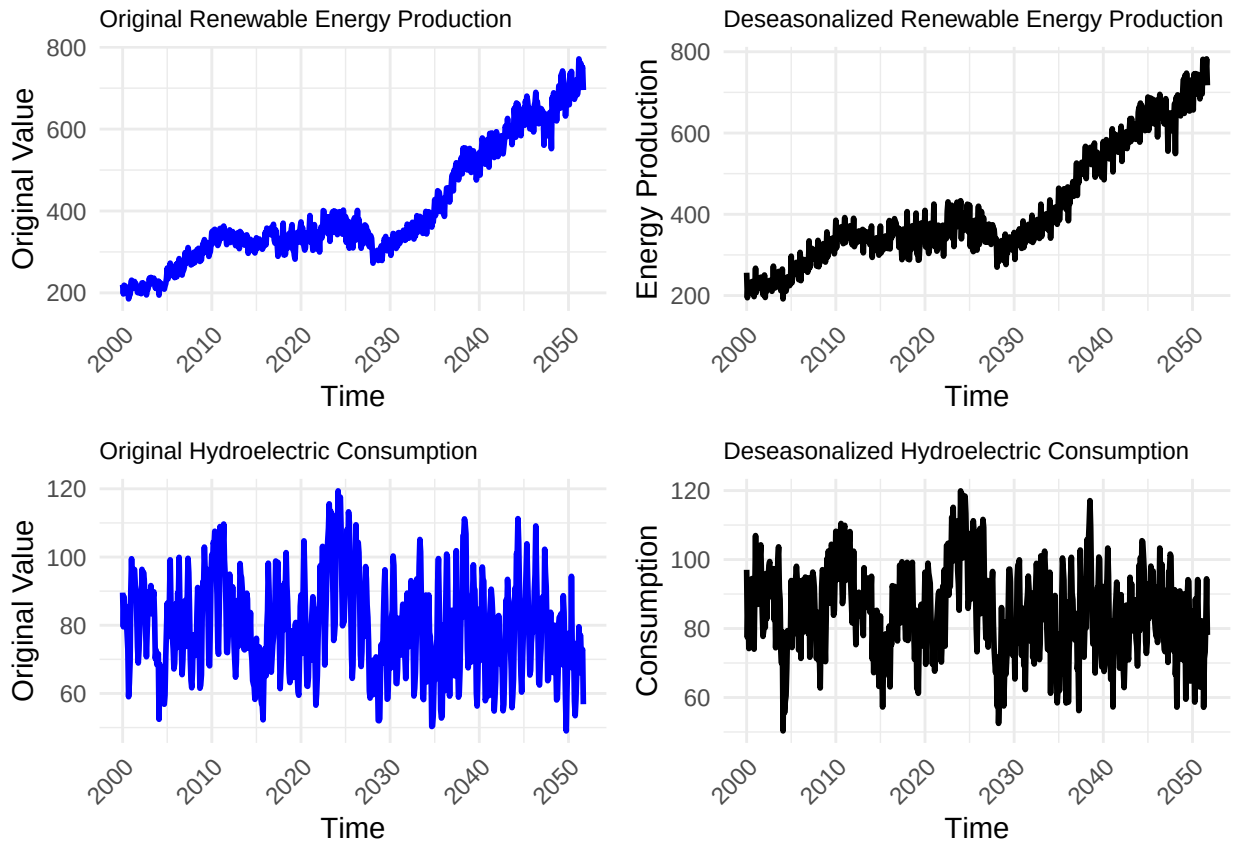
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```r
# Create plots for Hydroelectric Consumption
hydro_deseasoned_plot <- ggplot(hydro_deseasoned_df, aes(x = Date)) +
  geom_line(aes(y = Deseasonalized), size = 1) +
  ggtitle("Deseasonalized Hydroelectric Consumption") +
  xlab("Time") + ylab("Consumption") +
  theme_minimal()+
    theme(
      plot.title = element_text(size = 9),
      axis.text.x = element_text(angle = 45, hjust = 1))

# Compare with original time series
renewable_original_plot <- ggplot(data.frame(Date = time(renewable_ts),
                                             Value = as.numeric(renewable_ts)),
                                  aes(x = Date, y = Value)) +
  geom_line(color = "blue", size = 1) +
  ggtitle("Original Renewable Energy Production") +
  xlab("Time") + ylab("Original Value") +
  theme_minimal()+
    theme(
      plot.title = element_text(size = 9),
      axis.text.x = element_text(angle = 45, hjust = 1))


hydro_original_plot <- ggplot(data.frame(Date = time(hydro_ts),
                                         Value = as.numeric(hydro_ts)),
                              aes(x = Date, y = Value)) +
  geom_line(color = "blue", size = 1) +
  ggtitle("Original Hydroelectric Consumption") +
  xlab("Time") + ylab("Original Value") +
  theme_minimal()+
    theme(
      plot.title = element_text(size = 9),
      axis.text.x = element_text(angle = 45, hjust = 1))
```

```
# Arrange all plots for comparison
final_comparison8 <- plot_grid(
  renewable_original_plot, renewable_deseasoned_plot,
  hydro_original_plot, hydro_deseasoned_plot,
  nrow = 2, ncol = 2
)
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```

```
# Display the final plot comparison
print(final_comparison8)
```

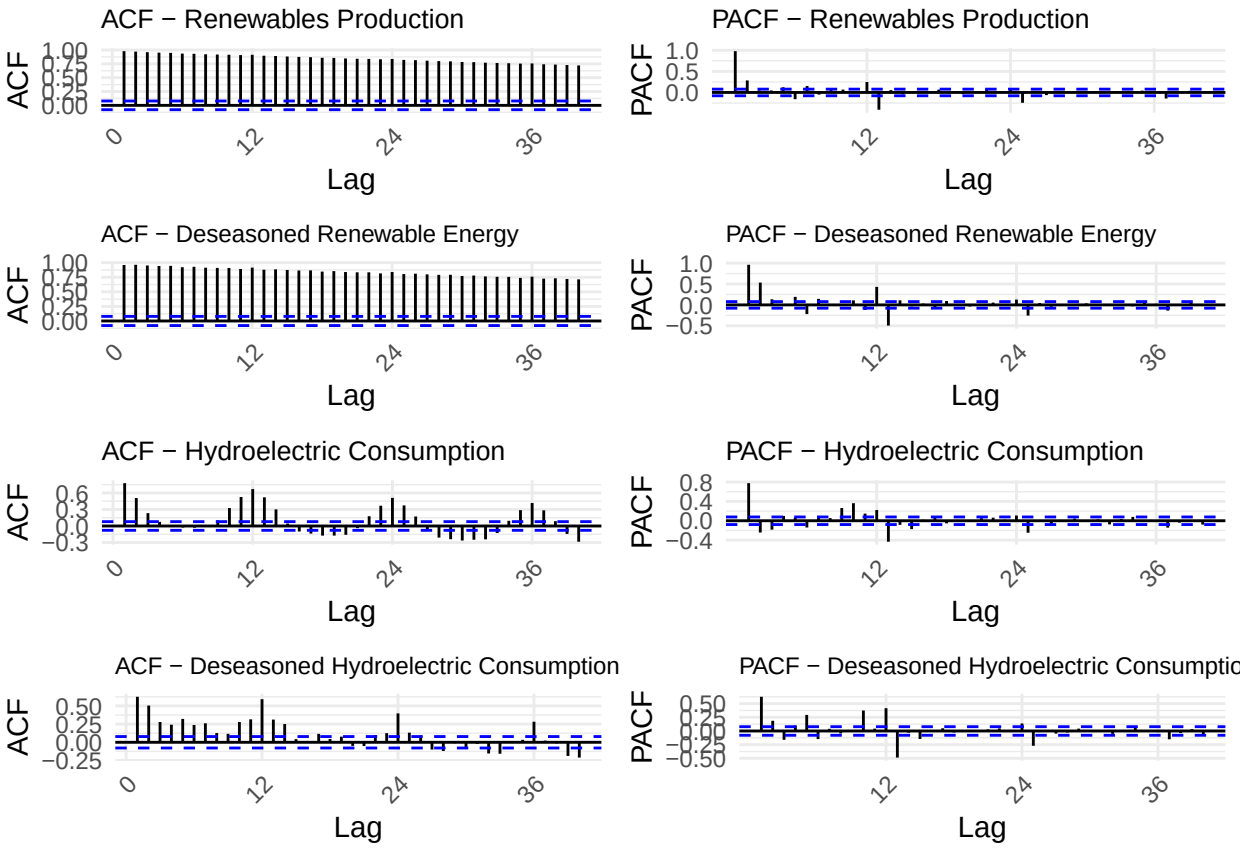


Comparison: Among the Renewable Energy Production and the Hydroelectric Power Consumption, there did not display a considerable amount change in the overall trends (with some minor change in fluctions within short time frames) comparing the original ts plots to their deseasoned plots, which could imply a lack of seaonality. This could also imply a considerable amount of randomness and irregularities.

Q9

Plot ACF and PACF for the deseason series and compare with the plots from Q1. You may use plot_grid() again to get them side by side, but not mandatory. Did the plots change? How?

```
# Function to create ACF and PACF plots
plot_acf_pacf <- function(ts_data, ts_name) {
  acf_plot <- ggAcf(ts_data, lag.max = 40) +
    ggtitle(paste("ACF -", ts_name)) +
    theme_minimal()+
    theme(
      plot.title = element_text(size = 9),
      axis.text.x = element_text(angle = 45, hjust = 1))

  pacf_plot <- ggPacf(ts_data, lag.max = 40) +
    ggtitle(paste("PACF -", ts_name)) +
    theme_minimal()+
    theme(
      plot.title = element_text(size = 9),
      axis.text.x = element_text(angle = 45, hjust = 1))

  return(list(acf_plot, pacf_plot))}

# Generate ACF and PACF plots for deseasoned series
renewable_deseasoned_plots <- plot_acf_pacf(ts_deseason_renewable,
                                            "Deseasoned Renewable Energy")
hydro_deseasoned_plots <- plot_acf_pacf(ts_deseason_hydro,
                                        "Deseasoned Hydroelectric Consumption")

# Compare original and deseasoned ACF/PACF plots using plot_grid
final_comparison9 <- plot_grid(
  # ACF and PACF original renewables
  renewable_plots[[2]],  renewable_plots[[3]],
  # Deseasoned renewables
  renewable_deseasoned_plots[[1]], renewable_deseasoned_plots[[2]],
  #Hydro ACF and PACF original
  hydro_plots[[2]], hydro_plots[[3]],
  # ACF and PACF for deseasoned hydro
  hydro_deseasoned_plots[[1]], hydro_deseasoned_plots[[2]],
  nrow = 4)

print(final_comparison9)
```



Answer: Comparing the orginal ACF and PACF plots to deseasoned plots, there seems to be some changes, but overall not very dramatic. For Renewables Production, the ACF ad PACF displayed slightly greater pattern of seasonality or more consistent and greater fluctuations in a 12-month interval. In terms of Hydroelectric Consumption, the deseasonsed PACF plot shows a more negative values, which means higher values in the past are associated with lower values in the present, while showing a similar seasonal pattern as the original ones. The ACF plot instead shows lower value after removing the seasonal component. This may reflect noise or irregularities in Hydroelectric consumption trend, although the ACF patterns overall shows spme visible seasonality overtime.