ENV 790.30 - Time Series Analysis for Energy Data | Spring 2026

Assignment 3 - Due date 02/03/26

Trudy

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 2025 Monthly Energy Review. This time 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(readxl)
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
```

```
library(forecast)
```

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

```
library(tseries)
library(Kendall)

file_path <- "./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx"

# Import raw energy data
raw_energy_ts <- read_excel(
  path = file_path,
  skip = 12,
  sheet = "Monthly Data",
  col_names = FALSE
)
```

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

```
# Extract header row
header_row <- read_excel(
  path = file_path,
  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 column names
colnames(raw_energy_ts) <- as.character(header_row[1, ])

# Construct data frame with required time series only
renewable_hydro_df <- raw_energy_ts %>%
  select(
    Date = 1,
    `Total Renewable Energy Production`,
    `Hydroelectric Power Consumption`
  )

# Preview
head(renewable_hydro_df)
```

```
## # A tibble: 6 x 3
##   Date                `Total Renewable Energy Production` Hydroelectric Power ~1
##   <dttm>                                            <dbl>                  <dbl>
## 1 1973-01-01 00:00:00                                220.                   89.6
## 2 1973-02-01 00:00:00                                197.                   79.5
## 3 1973-03-01 00:00:00                                219.                   88.3
## 4 1973-04-01 00:00:00                                209.                   83.2
## 5 1973-05-01 00:00:00                                216.                   85.6
## 6 1973-06-01 00:00:00                                208.                   82.1
## # i abbreviated name: 1: `Hydroelectric Power Consumption`
```

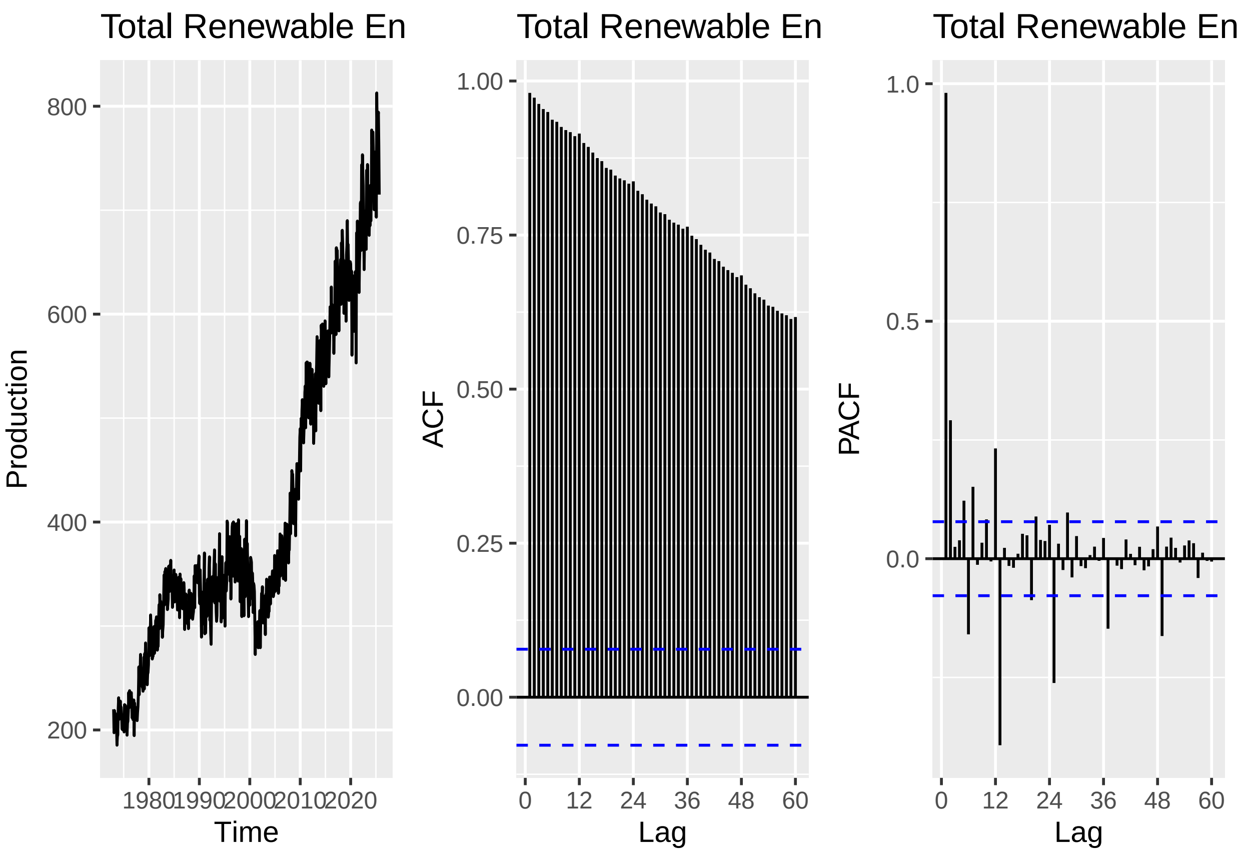
##Trend Component

Q1

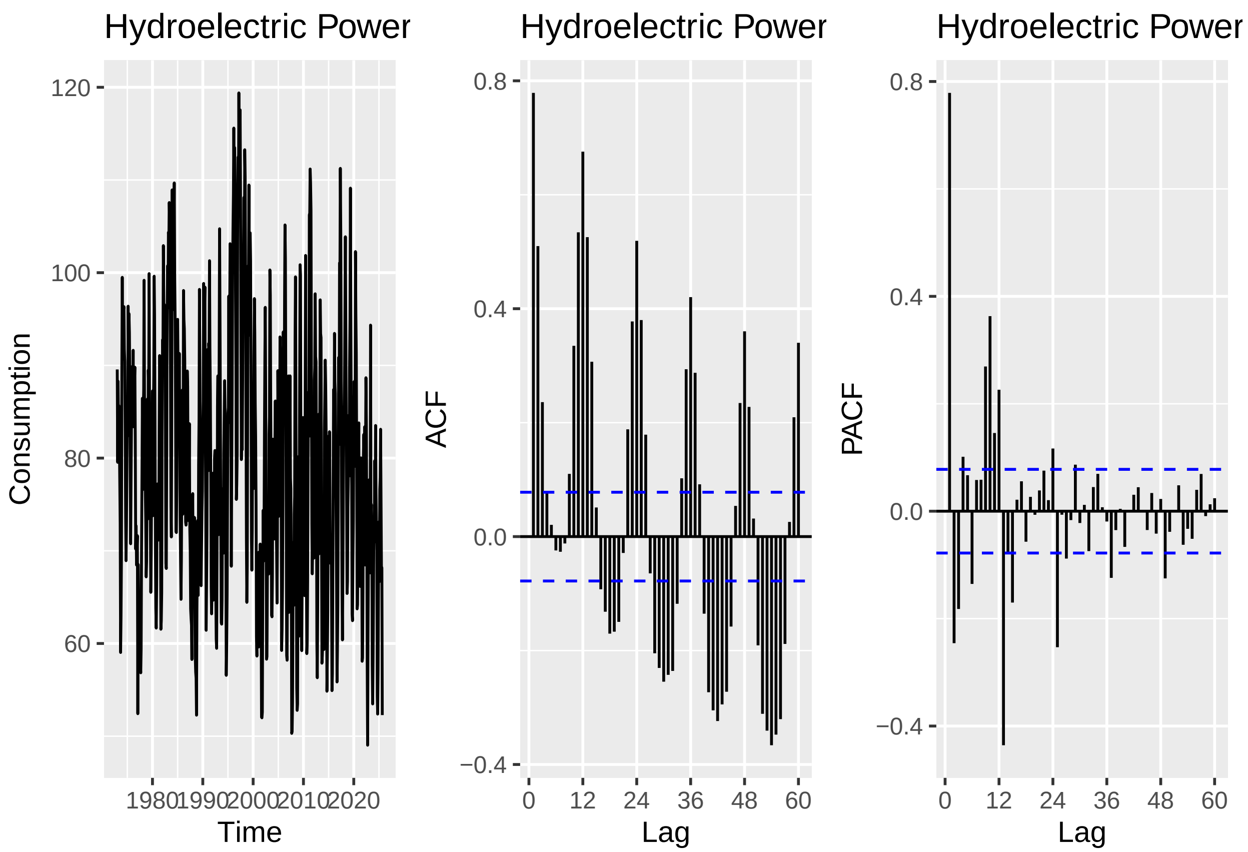
For each series (Total Renewable Production and Hydroelectric Consumption) create three plots arranged in a row (side-by-side): (1) time series plot, (2) ACF, (3) PACF. Use cowplot::plot_grid() to place them in a grid.

```
library(dplyr)
library(ggplot2)
library(cowplot)

#Create ts objects (monthly)
start_year  <- as.integer(format(min(renewable_hydro_df$Date), "%Y"))
start_month <- as.integer(format(min(renewable_hydro_df$Date), "%m"))
```

```r
ts_total_renew <- ts(
  renewable_hydro_df$`Total Renewable Energy Production`,
  start = c(start_year, start_month),
  frequency = 12
)

ts_hydro_cons <- ts(
  renewable_hydro_df$`Hydroelectric Power Consumption`,
  start = c(start_year, start_month),
  frequency = 12
)

#3 plots in a row using cowplot
make_row_plots <- function(ts_obj, series_name, y_label) {

  p_ts <- autoplot(ts_obj) +
    labs(title = paste0(series_name, " (Time Series)"),
         x = "Time", y = y_label)

  p_acf <- autoplot(Acf(ts_obj, lag.max = 60, plot = FALSE)) +
    labs(title = paste0(series_name, " (ACF)"),
         x = "Lag", y = "ACF")

  p_pacf <- autoplot(Pacf(ts_obj, lag.max = 60, plot = FALSE)) +
    labs(title = paste0(series_name, " (PACF)"),
         x = "Lag", y = "PACF")

  cowplot::plot_grid(p_ts, p_acf, p_pacf, nrow = 1, align = "hv")
}

#Row 1: Total Renewable Energy Production
q1_total_row <- make_row_plots(
  ts_obj = ts_total_renew,
  series_name = "Total Renewable Energy Production",
  y_label = "Production"
)

#Row 2: Hydroelectric Power Consumption
q1_hydro_row <- make_row_plots(
  ts_obj = ts_hydro_cons,
  series_name = "Hydroelectric Power Consumption",
  y_label = "Consumption"
)

# Print
q1_total_row
```


Total Renewable En
Production
800
600
400
200
1980
1990
2000
2010
2020
Time
Total Renewable En
ACF
1.00
0.75
0.50
0.25
0.00
0
12
24
36
48
60
Lag
Total Renewable En
PACF
1.0
0.5
0.0
0
12
24
36
48
60
Lag

```
q1_hydro_row
```



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: Total Renewable Energy Production shows a strong upward long-term trend, as indicated by the increasing pattern in the time series plot and the slowly decaying ACF, suggesting a non-stationary series with trend. In contrast, Hydroelectric Power Consumption does not display a clear long-term trend and instead fluctuates around a relatively stable mean, indicating a roughly stationary series without a pronounced trend component.

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.

```
# Create time index
t <- 1:nrow(renewable_hydro_df)

# Linear trend: Total Renewable Energy Production
lm_total_renew <- lm(
  renewable_hydro_df$`Total Renewable Energy Production` ~ t
)
summary(lm_total_renew)
```

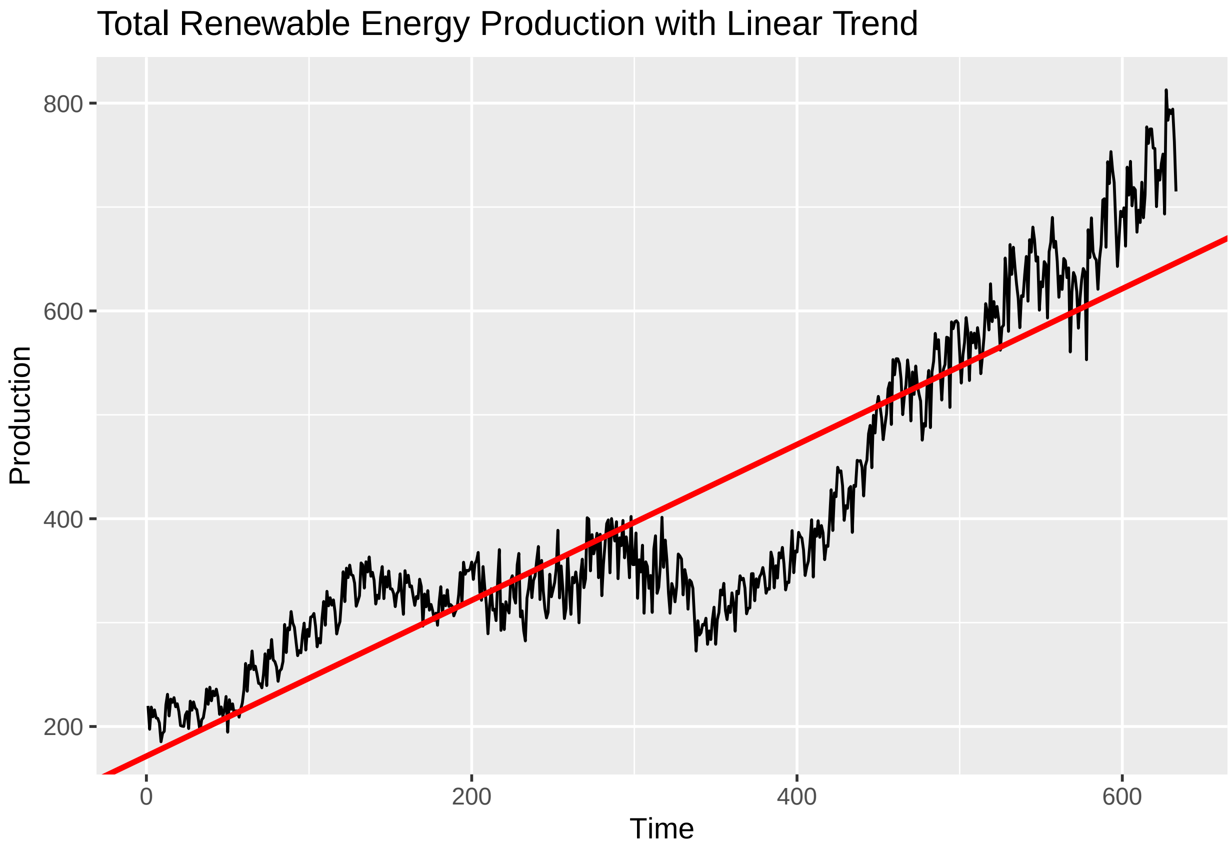
```
##
## Call:
## lm(formula = renewable_hydro_df$‘Total Renewable Energy Production‘ ~
##     t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -154.81  -39.55   12.52   41.49  171.15
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 171.44868    5.11085   33.55   <2e-16 ***
## t             0.74999    0.01397   53.69   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 64.22 on 631 degrees of freedom
## Multiple R-squared:  0.8204, Adjusted R-squared:  0.8201
## F-statistic:  2883 on 1 and 631 DF,  p-value: < 2.2e-16
```

```r
# Store coefficients
beta0_total <- as.numeric(lm_total_renew$coefficients[1])  # intercept
beta1_total <- as.numeric(lm_total_renew$coefficients[2])  # slope

# Linear trend: Hydroelectric Power Consumption
lm_hydro_cons <- lm(
  renewable_hydro_df$`Hydroelectric Power Consumption` ~ t
)
summary(lm_hydro_cons)
```

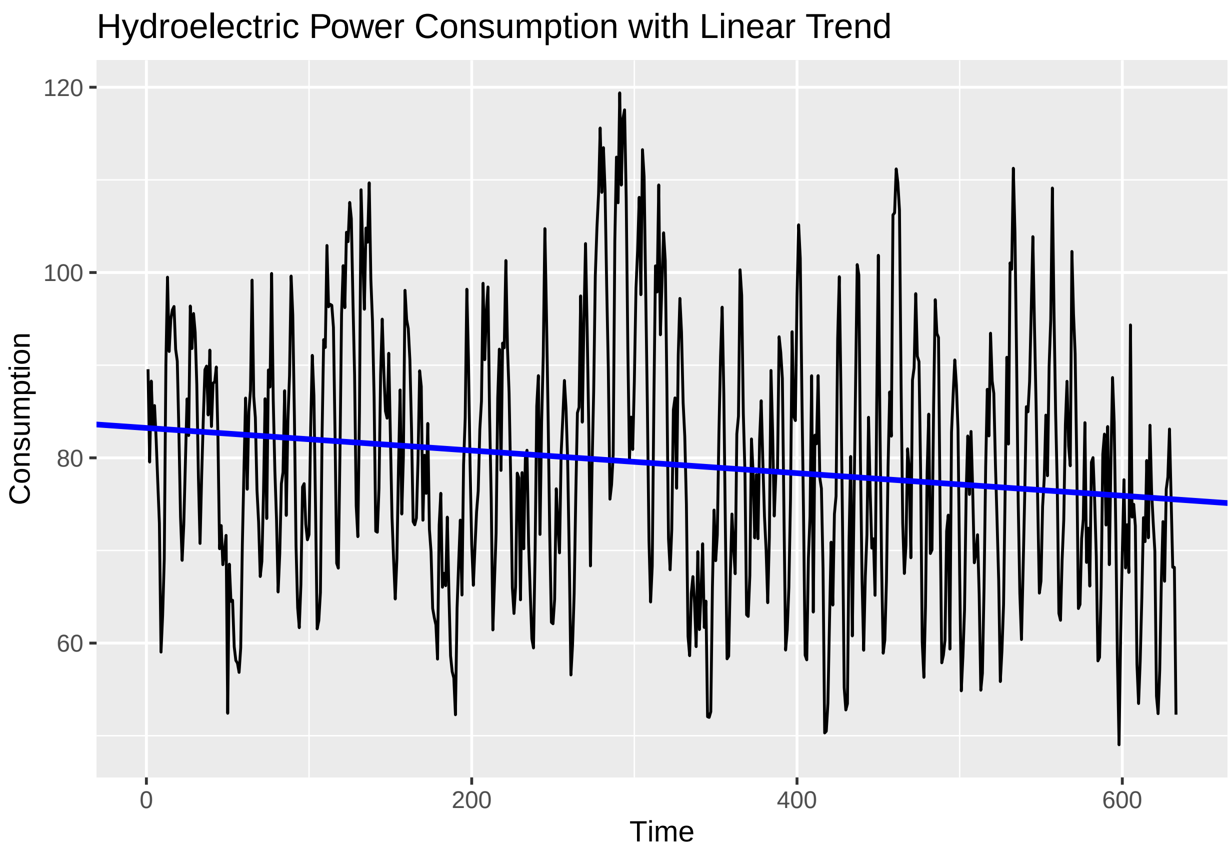
```
##
## Call:
## lm(formula = renewable_hydro_df$‘Hydroelectric Power Consumption‘ ~
##     t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.190 -10.214  -0.715   8.909  39.723
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 83.223802   1.110552  74.939  < 2e-16 ***
## t           -0.012199   0.003035  -4.019 6.55e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 13.95 on 631 degrees of freedom
## Multiple R-squared:  0.02496,    Adjusted R-squared:  0.02342
## F-statistic: 16.15 on 1 and 631 DF,  p-value: 6.547e-05
```

```r
# Store coefficients
beta0_hydro <- as.numeric(lm_hydro_cons$coefficients[1])
beta1_hydro <- as.numeric(lm_hydro_cons$coefficients[2])
```

```r
# ---- PLOTS WITH TREND LINES ----

library(ggplot2)

# Total Renewable plot with trend
ggplot(renewable_hydro_df, aes(x = t,
       y = `Total Renewable Energy Production`)) +
  geom_line() +
  geom_abline(intercept = beta0_total,
              slope = beta1_total,
              color = "red", linewidth = 1) +
  labs(title = "Total Renewable Energy Production with Linear Trend",
       x = "Time",
       y = "Production")
```



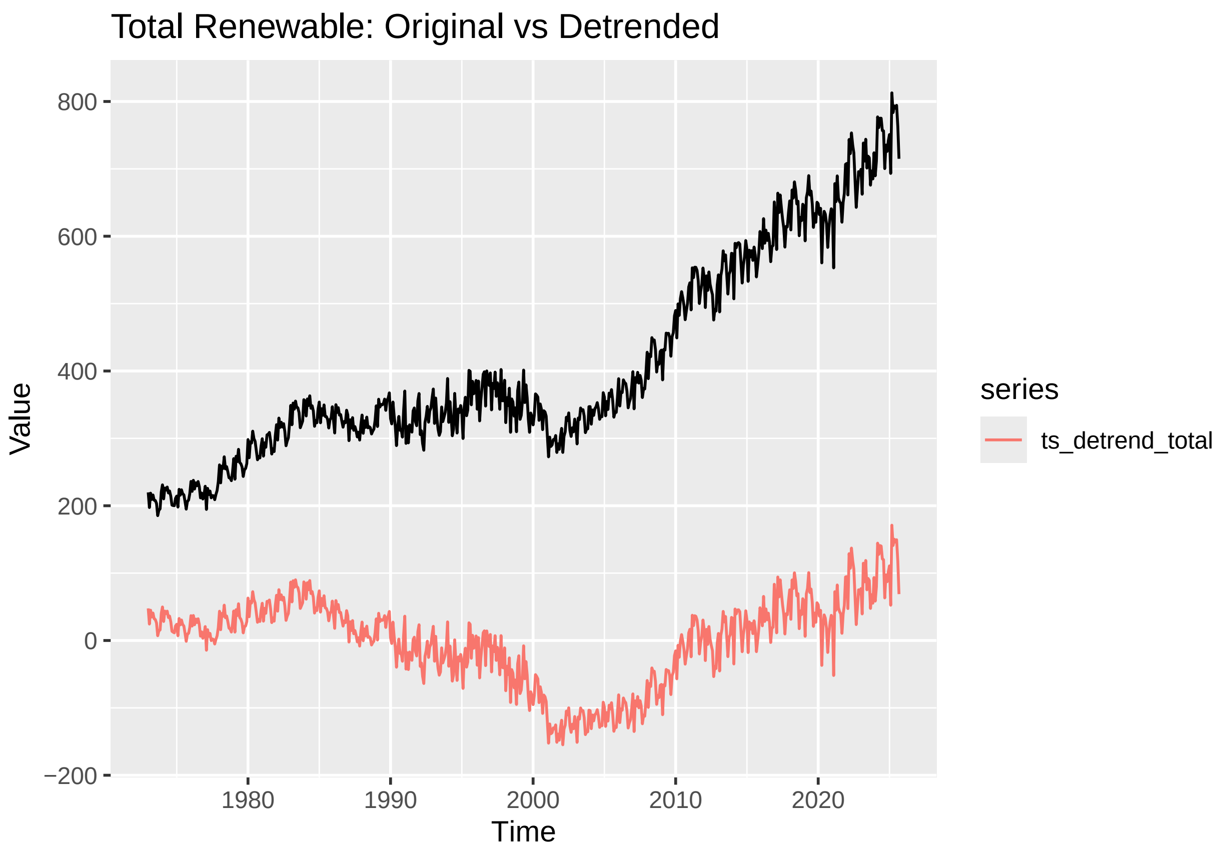
```r
# Hydro plot with trend
ggplot(renewable_hydro_df, aes(x = t,
       y = `Hydroelectric Power Consumption`)) +
  geom_line() +
  geom_abline(intercept = beta0_hydro,
              slope = beta1_hydro,
              color = "blue", linewidth = 1) +
  labs(title = "Hydroelectric Power Consumption with Linear Trend",
       x = "Time",
       y = "Consumption")
```



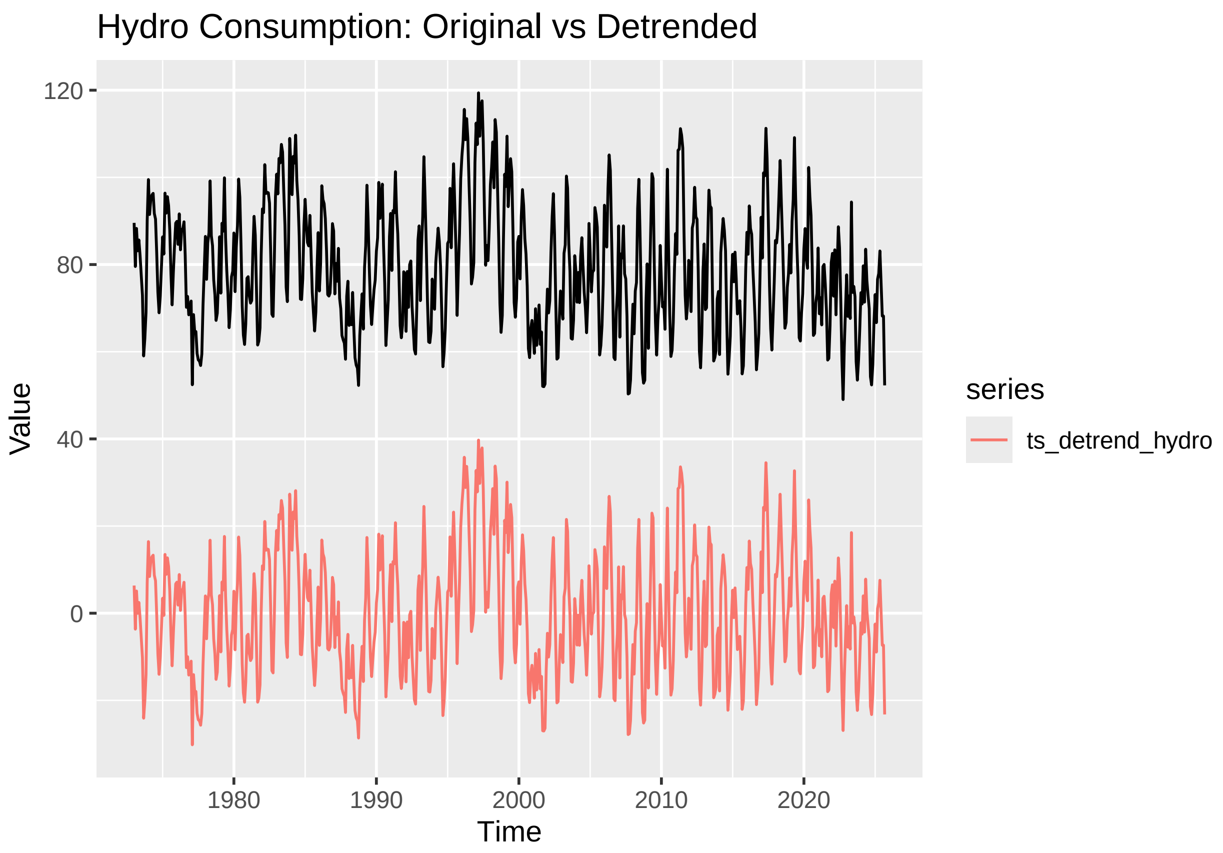
Q4

Use the regression coefficients to detrend each series (subtract fitted linear trend). Plot detrended series and compare with the original time series from Q1. Describe what changed.

```
# 1) Detrend (y_t - fitted trend)
detrend_total <- renewable_hydro_df$`Total Renewable Energy Production` -
  (beta0_total + beta1_total * t)

detrend_hydro <- renewable_hydro_df$`Hydroelectric Power Consumption` -
  (beta0_hydro + beta1_hydro * t)

# 2) Convert detrended vectors into ts objects (same start/frequency as original ts)
ts_detrend_total <- ts(detrend_total, start = start(ts_total_renew), frequency = frequency(ts_total_rene
ts_detrend_hydro <- ts(detrend_hydro, start = start(ts_hydro_cons),  frequency = frequency(ts_hydro_cons

# 3) Plot: original vs detrended (overlay)
autoplot(ts_total_renew) +
  autolayer(ts_detrend_total) +
  labs(title = "Total Renewable: Original vs Detrended", x = "Time", y = "Value")
```



```
autoplot(ts_hydro_cons) +
  autolayer(ts_detrend_hydro) +
  labs(title = "Hydro Consumption: Original vs Detrended", x = "Time", y = "Value")
```



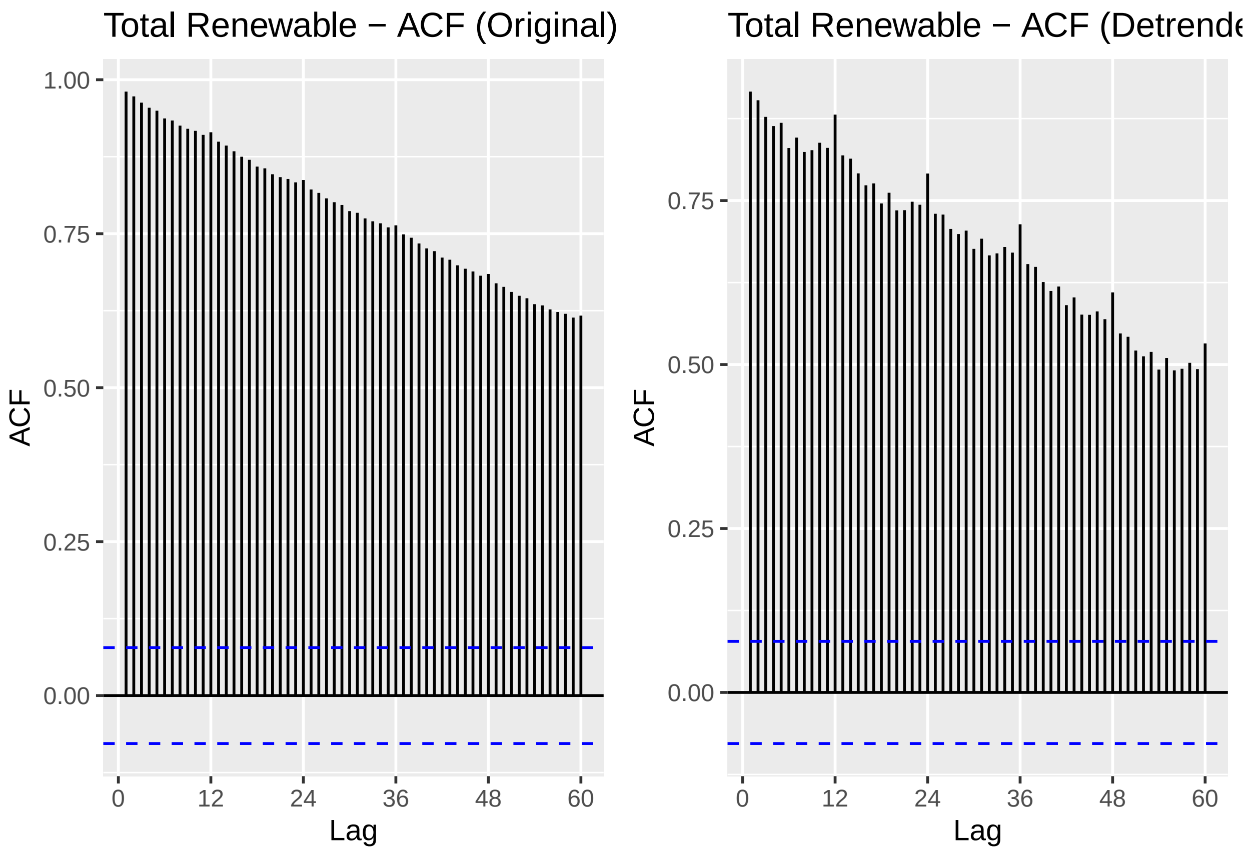
> Answer: After detrending, the strong upward trend in total renewable energy production is removed. The detrended series fluctuates around a relatively constant level, indicating that the long-term growth component has been successfully eliminated and only short-term variation remains. For hydroelectric power consumption, detrending produces only a minor change. Since the original series does not exhibit a strong linear trend, the detrended series remains similar to the original and continues to fluctuate around a stable mean.

Q5

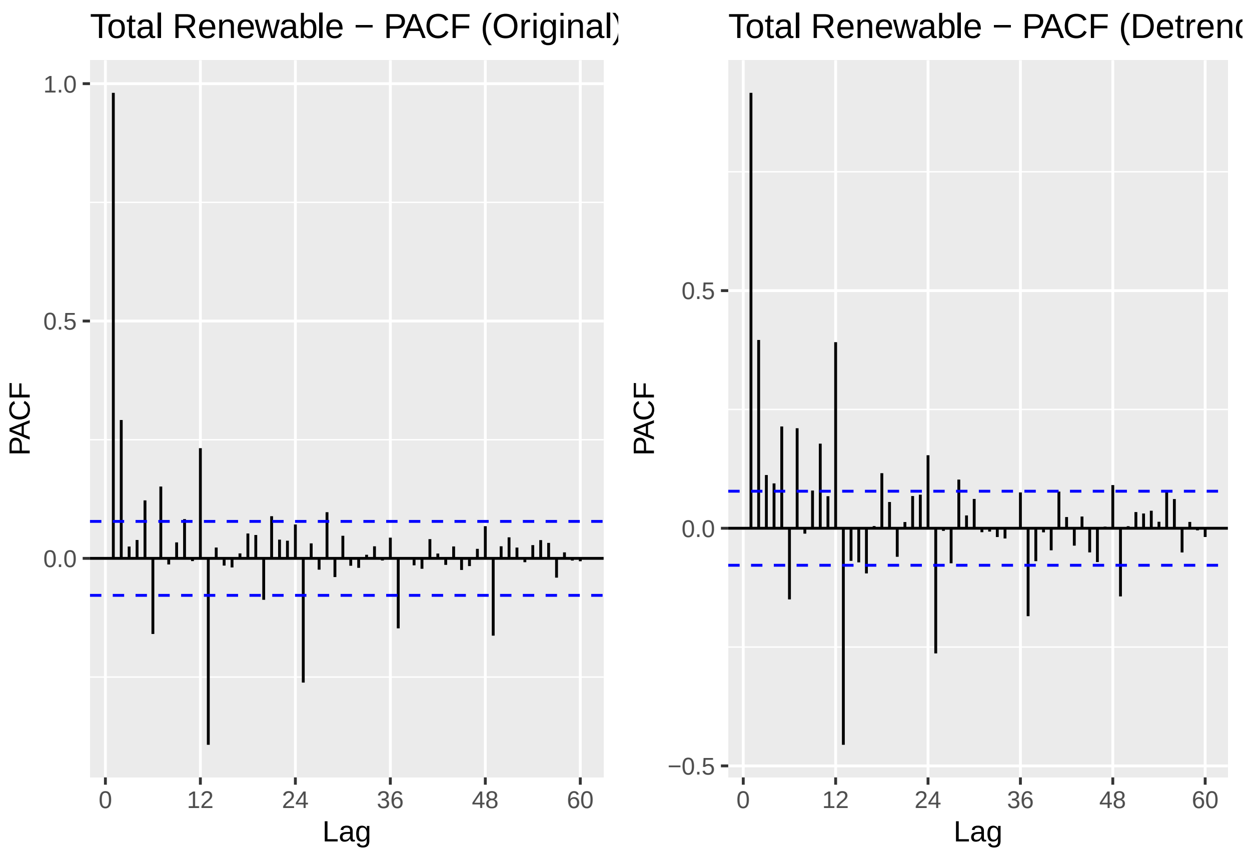
Plot ACF and PACF for the detrended series and compare with the plots from Q1. You may use plot_grid() again to get them side by side to make it easier to compare. Did the plots change? How?

```
#Total Renewable
p_total_acf_orig  <- autoplot(Acf(ts_total_renew, lag.max = 60, plot = FALSE)) +
  labs(title = "Total Renewable - ACF (Original)")

p_total_acf_det   <- autoplot(Acf(ts_detrend_total, lag.max = 60, plot = FALSE)) +
  labs(title = "Total Renewable - ACF (Detrended)")

p_total_pacf_orig <- autoplot(Pacf(ts_total_renew, lag.max = 60, plot = FALSE)) +
  labs(title = "Total Renewable - PACF (Original)")

p_total_pacf_det  <- autoplot(Pacf(ts_detrend_total, lag.max = 60, plot = FALSE)) +
  labs(title = "Total Renewable - PACF (Detrended)")

# ACF comparison
```

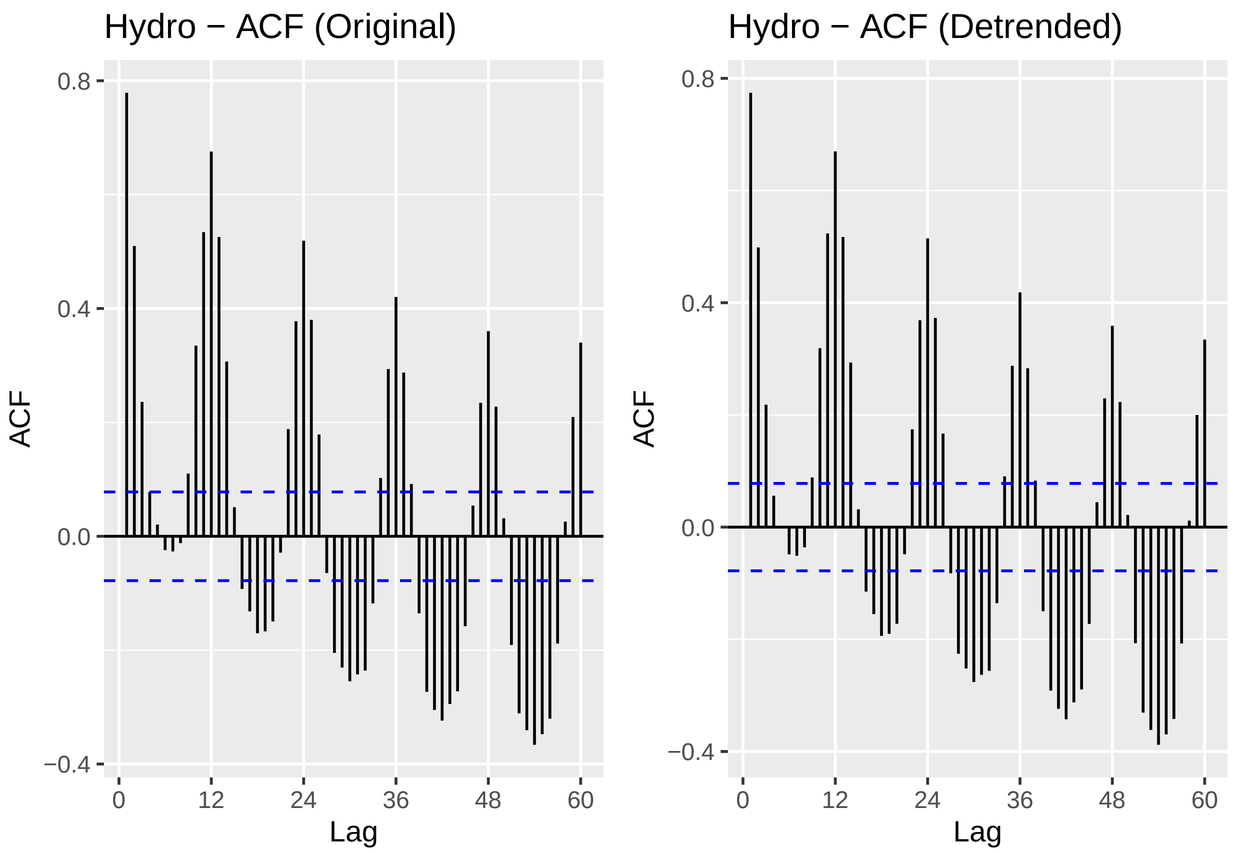
```
plot_grid(p_total_acf_orig, p_total_acf_det, nrow = 1)
```



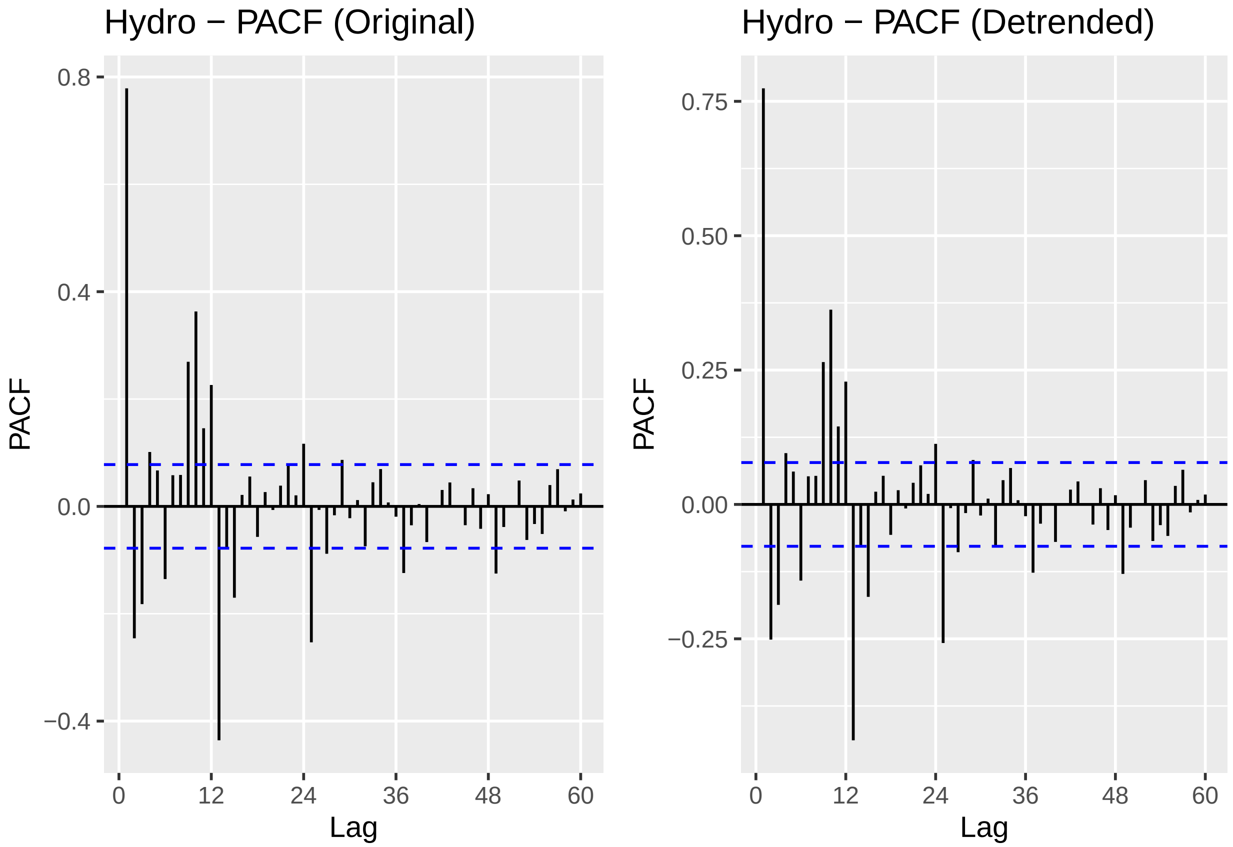
```
# PACF comparison
plot_grid(p_total_pacf_orig, p_total_pacf_det, nrow = 1)
```



```r
#Hydroelectric
p_hydro_acf_orig  <- autoplot(Acf(ts_hydro_cons, lag.max = 60, plot = FALSE)) +
  labs(title = "Hydro - ACF (Original)")

p_hydro_acf_det   <- autoplot(Acf(ts_detrend_hydro, lag.max = 60, plot = FALSE)) +
  labs(title = "Hydro - ACF (Detrended)")

p_hydro_pacf_orig <- autoplot(Pacf(ts_hydro_cons, lag.max = 60, plot = FALSE)) +
  labs(title = "Hydro - PACF (Original)")

p_hydro_pacf_det  <- autoplot(Pacf(ts_detrend_hydro, lag.max = 60, plot = FALSE)) +
  labs(title = "Hydro - PACF (Detrended)")

# ACF comparison
plot_grid(p_hydro_acf_orig, p_hydro_acf_det, nrow = 1)
```



```
# PACF comparison
plot_grid(p_hydro_pacf_orig, p_hydro_pacf_det, nrow = 1)
```



> Answer: For total renewable energy production, the ACF and PACF change substantially after detrending. The ACF decays faster and shows lower overall correlation, indicating that much of the strong persistence in the original series was driven by the trend component. The PACF is no longer dominated by a single large first lag, suggesting a simpler short-term correlation structure. For hydroelectric power consumption, the ACF and PACF plots change very little after detrending. The overall correlation patterns remain similar, which is consistent with the fact that the original series does not exhibit a strong linear trend.

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: Total renewable energy production does not show a clear seasonal pattern, as the series is dominated by a long-term upward trend and the ACF does not exhibit strong peaks at seasonal lags. In contrast, hydroelectric power consumption displays clear seasonality, with regular cyclical fluctuations over time and significant spikes in the ACF at lags of 12 and its multiples, indicating an annual seasonal pattern.

Q7

Use function *lm()* to fit a seasonal means model (i.e. using the seasonal dummies) to 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?

```
# Total Renewable (ORIGINAL)
d_total <- seasonaldummy(ts_total_renew)             # 11 monthly dummies (baseline month omitted)
season_model_total <- lm(ts_total_renew ~ d_total)
summary(season_model_total)
```

```
##
## Call:
## lm(formula = ts_total_renew ~ d_total)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -213.33  -97.36  -59.88  121.55  389.62
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  417.265     21.096  19.779   <2e-16 ***
## d_totalJan     2.090     29.693   0.070    0.944
## d_totalFeb   -34.524     29.693  -1.163    0.245
## d_totalMar     5.956     29.693   0.201    0.841
## d_totalApr    -6.900     29.693  -0.232    0.816
## d_totalMay     8.162     29.693   0.275    0.784
## d_totalJun    -2.231     29.693  -0.075    0.940
## d_totalJul     3.864     29.693   0.130    0.897
## d_totalAug    -3.978     29.693  -0.134    0.893
## d_totalSep   -29.033     29.693  -0.978    0.329
## d_totalOct   -19.937     29.834  -0.668    0.504
## d_totalNov   -20.617     29.834  -0.691    0.490
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 152.1 on 621 degrees of freedom
## Multiple R-squared:  0.008243,   Adjusted R-squared:  -0.009324
## F-statistic: 0.4692 on 11 and 621 DF,  p-value: 0.9223
```

```
# store coefficients
alpha_total <- as.numeric(season_model_total$coefficients[1])      # intercept
gamma_total <- as.numeric(season_model_total$coefficients[2:12])   # seasonal dummy coefficients

# Hydro (ORIGINAL)
d_hydro <- seasonaldummy(ts_hydro_cons)
season_model_hydro <- lm(ts_hydro_cons ~ d_hydro)
summary(season_model_hydro)
```

```
##
## Call:
## lm(formula = ts_hydro_cons ~ d_hydro)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -30.895  -6.368  -0.595   6.213  32.557 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   79.724      1.436  55.511  < 2e-16 ***
## d_hydroJan     4.951      2.021   2.449 0.014591 *  
## d_hydroFeb    -2.415      2.021  -1.195 0.232608    
## d_hydroMar     7.116      2.021   3.520 0.000463 ***
## d_hydroApr     5.614      2.021   2.777 0.005649 ** 
## d_hydroMay    14.080      2.021   6.965 8.38e-12 ***
## d_hydroJun    10.780      2.021   5.333 1.36e-07 ***
## d_hydroJul     4.003      2.021   1.980 0.048091 *  
## d_hydroAug    -5.320      2.021  -2.632 0.008710 ** 
## d_hydroSep   -16.598      2.021  -8.211 1.28e-15 ***
## d_hydroOct   -16.329      2.031  -8.040 4.56e-15 ***
## d_hydroNov   -10.782      2.031  -5.308 1.54e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 10.36 on 621 degrees of freedom
## Multiple R-squared:  0.4714, Adjusted R-squared:  0.4621 
## F-statistic: 50.35 on 11 and 621 DF,  p-value: < 2.2e-16
```

```
alpha_hydro <- as.numeric(season_model_hydro$coefficients[1])
gamma_hydro <- as.numeric(season_model_hydro$coefficients[2:12])
```

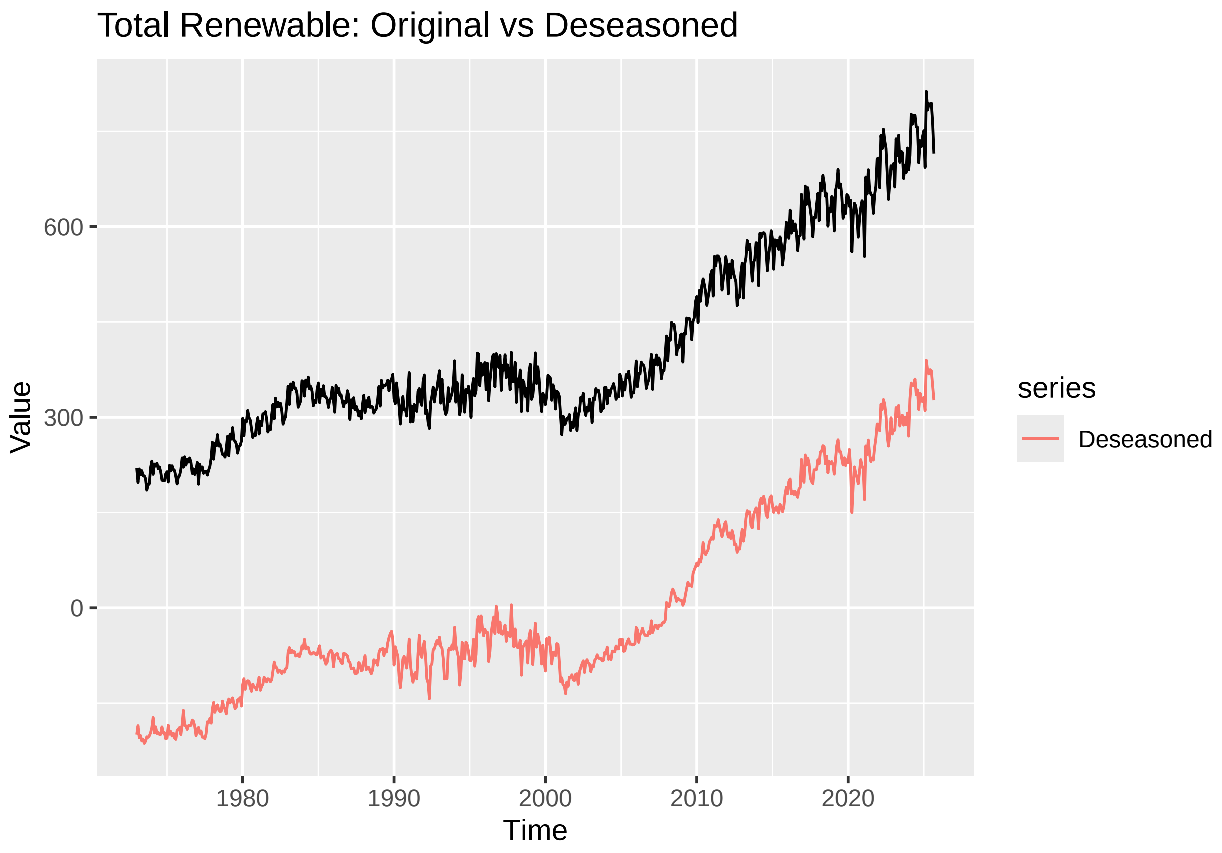
> Answer: The seasonal means regression shows that Total Renewable Energy Production does not exhibit a significant seasonal pattern, as none of the monthly dummies are significant and the overall model is not significant. In contrast, Hydroelectric Power Consumption shows strong seasonality, with many monthly dummies statistically significant and a high r squared (about 0.47). These results match the visual impression from Q6.

Q8

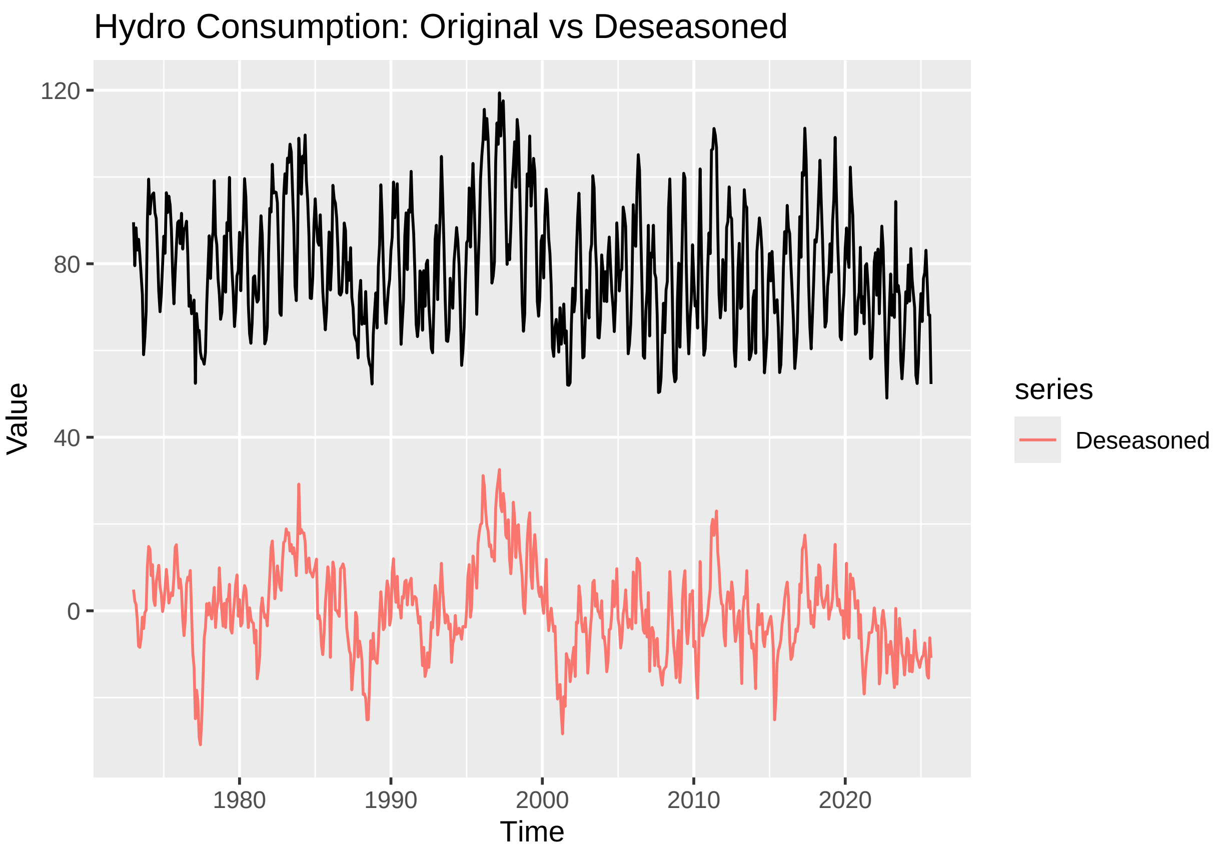
Use the regression coefficients from Q7 to deseason the series. Plot the deseason series and compare with the plots from part Q1. Did anything change?

```
# Total: compute fitted seasonal component
season_comp_total <- fitted(season_model_total)

# remove seasonality (deseasoned series)
deseason_total <- as.numeric(ts_total_renew) - season_comp_total
ts_deseason_total <- ts(deseason_total, start = start(ts_total_renew), frequency = frequency(ts_total_r

# Hydro:  compute fitted seasonal component
season_comp_hydro <- fitted(season_model_hydro)

# remove seasonality (deseasoned series)
deseason_hydro <- as.numeric(ts_hydro_cons) - season_comp_hydro
ts_deseason_hydro <- ts(deseason_hydro, start = start(ts_hydro_cons), frequency = frequency(ts_hydro_co
```

```r
# plots: original vs deseasoned
autoplot(ts_total_renew) +
  autolayer(ts_deseason_total, series = "Deseasoned") +
  labs(title = "Total Renewable: Original vs Deseasoned", x = "Time", y = "Value")
```



```r
autoplot(ts_hydro_cons) +
  autolayer(ts_deseason_hydro, series = "Deseasoned") +
  labs(title = "Hydro Consumption: Original vs Deseasoned", x = "Time", y = "Value")
```

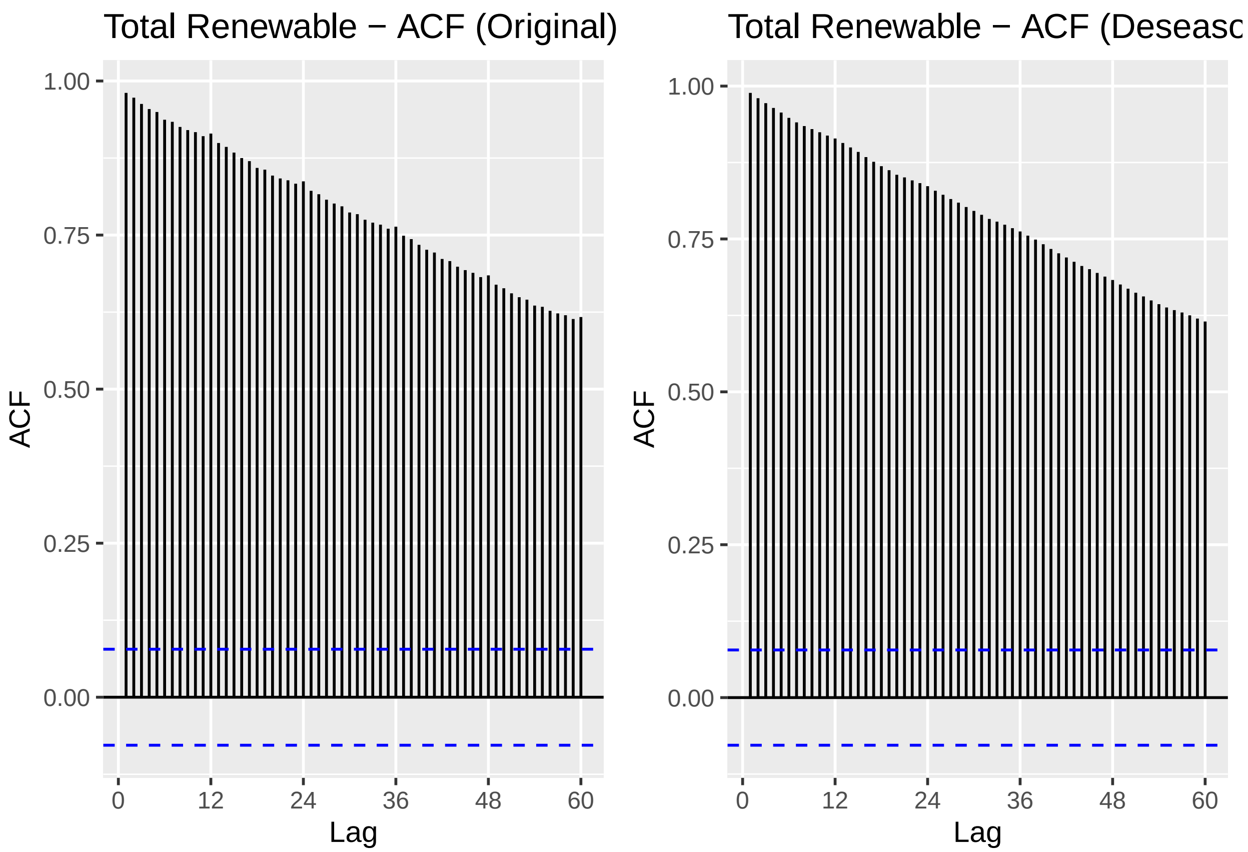


> Answer: For Total Renewable Energy Production, deseasoning does not lead to noticeable changes, confirming the absence of strong seasonal effects. For Hydroelectric Power Consumption, the regular annual oscillations are largely removed after deseasoning, and the series becomes more stationary around a stable mean. This confirms that hydro consumption contains a strong seasonal component.

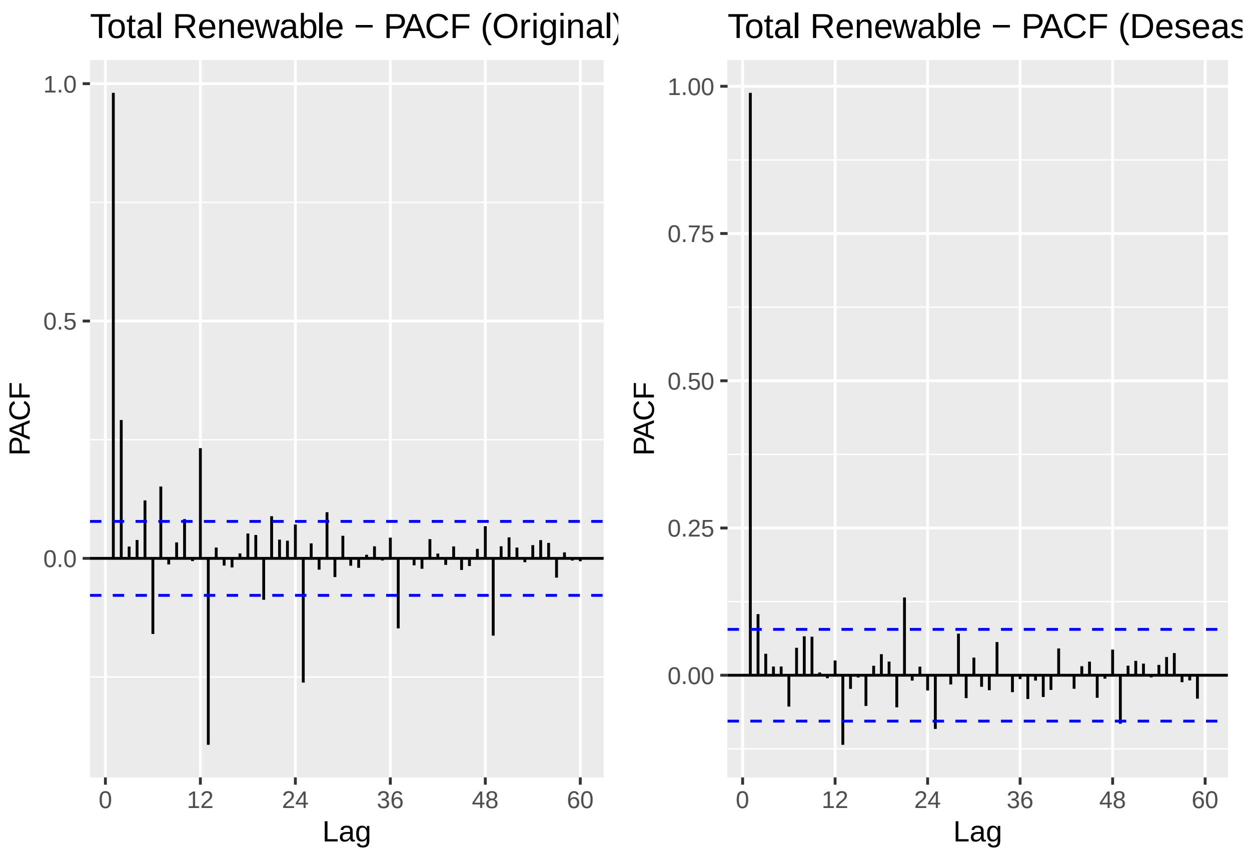
Q9

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. Did the plots change? How?

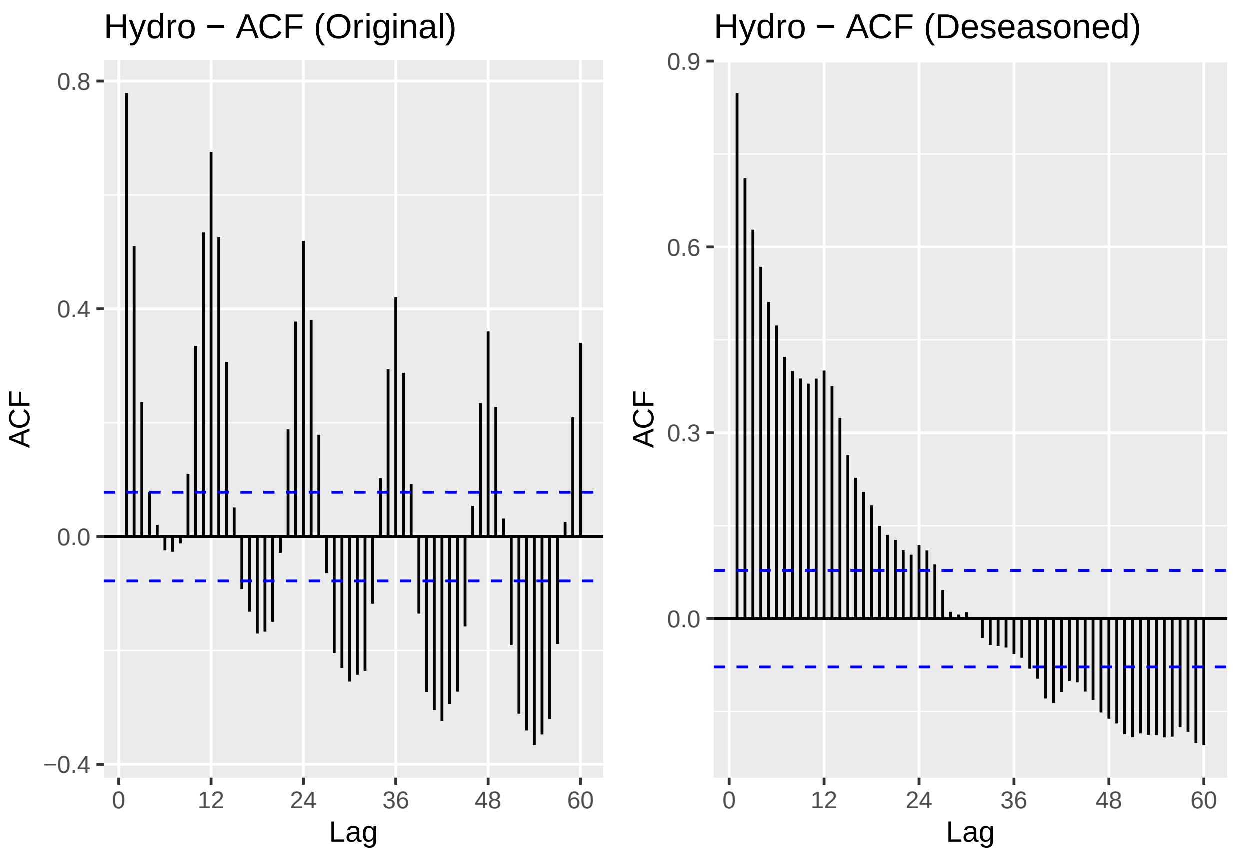
```
lag_max <- 60

#Total Renewable: ACF (Original vs Deseasoned)
p_total_acf_orig <- ggAcf(ts_total_renew, lag.max = lag_max) +
  labs(title = "Total Renewable - ACF (Original)", x = "Lag", y = "ACF")

p_total_acf_des <- ggAcf(ts_deseason_total, lag.max = lag_max) +
  labs(title = "Total Renewable - ACF (Deseasoned)", x = "Lag", y = "ACF")

plot_grid(p_total_acf_orig, p_total_acf_des, nrow = 1)
```



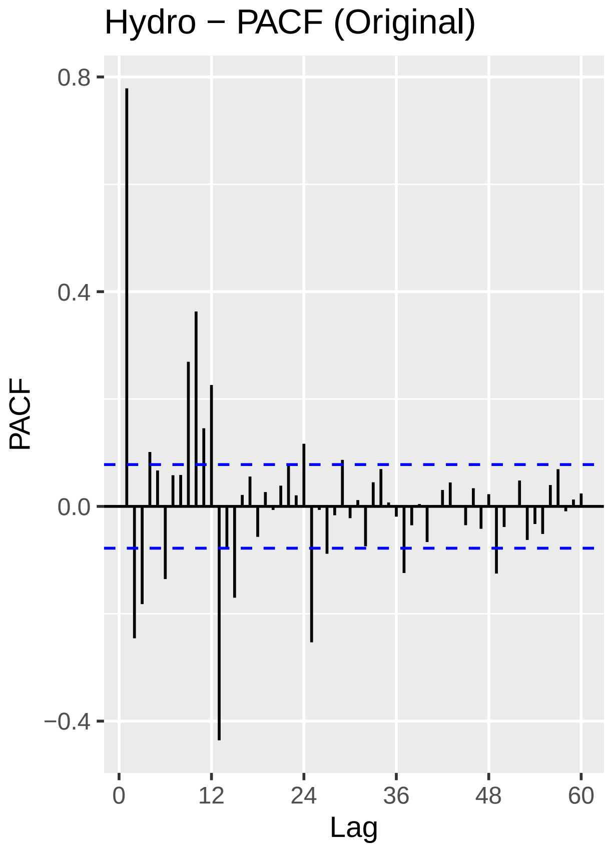
```
#Total Renewable: PACF (Original vs Deseasoned)
p_total_pacf_orig <- ggPacf(ts_total_renew, lag.max = lag_max) +
  labs(title = "Total Renewable - PACF (Original)", x = "Lag", y = "PACF")

p_total_pacf_des <- ggPacf(ts_deseason_total, lag.max = lag_max) +
  labs(title = "Total Renewable - PACF (Deseasoned)", x = "Lag", y = "PACF")

plot_grid(p_total_pacf_orig, p_total_pacf_des, nrow = 1)
```

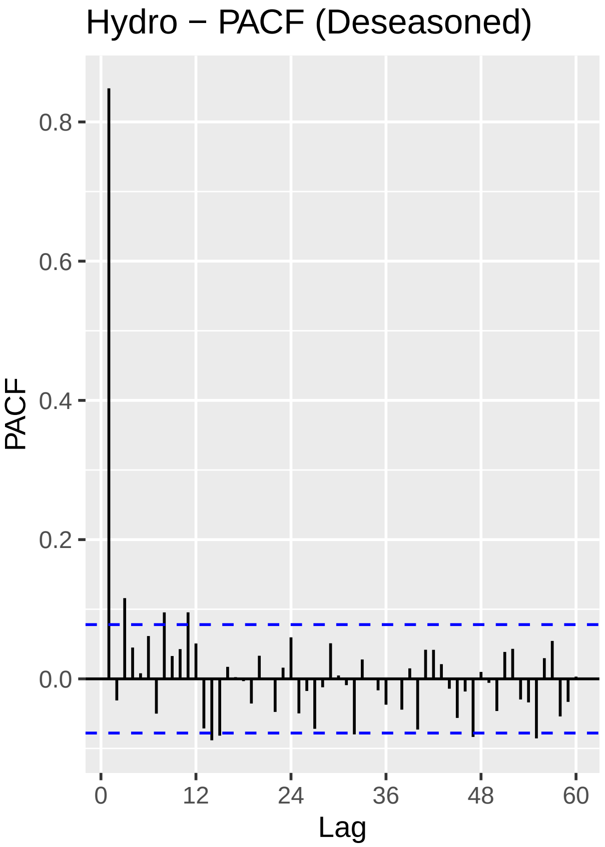


```
#Hydro: ACF (Original vs Deseasoned)
p_hydro_acf_orig <- ggAcf(ts_hydro_cons, lag.max = lag_max) +
  labs(title = "Hydro - ACF (Original)", x = "Lag", y = "ACF")

p_hydro_acf_des <- ggAcf(ts_deseason_hydro, lag.max = lag_max) +
  labs(title = "Hydro - ACF (Deseasoned)", x = "Lag", y = "ACF")

plot_grid(p_hydro_acf_orig, p_hydro_acf_des, nrow = 1)
```



```r
#Hydro: PACF (Original vs Deseasoned)
p_hydro_pacf_orig <- ggPacf(ts_hydro_cons, lag.max = lag_max) +
  labs(title = "Hydro - PACF (Original)", x = "Lag", y = "PACF")

p_hydro_pacf_des <- ggPacf(ts_deseason_hydro, lag.max = lag_max) +
  labs(title = "Hydro - PACF (Deseasoned)", x = "Lag", y = "PACF")

plot_grid(p_hydro_pacf_orig, p_hydro_pacf_des, nrow = 1)
```





> Answer: Yes, the plots changed, especially for the hydro series. For Total Renewable Energy Production, the ACF of the deseasoned series remains very high and decays slowly, similar to the original series. This indicates that removing seasonality does not significantly change the strong long-term persistence, and the series is still dominated by a trend component. The PACF also still shows a very large spike at lag 1, suggesting strong autocorrelation remains. For Hydroelectric Power Consumption, the changes are more substantial. In the original series, the ACF shows clear seasonal spikes at multiples of 12 lags, reflecting strong annual seasonality. After deseasoning, these seasonal spikes largely disappear, and the ACF decays more smoothly. The PACF of the deseasoned series also shows fewer significant spikes beyond lag 1. Overall, deseasoning successfully removes most of the seasonal structure in the hydro series, but has little effect on the total renewable series, which is mainly driven by long-term trend rather than seasonality.