

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

## Assignment 3 - Due date 02/03/26

Bella Huang

### 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\_Consumption.xlsx”. 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(forecast)
```

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

```
library(tseries)  
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(Kendall)
library(readxl)
library(openxlsx)
library(ggplot2)
library(cowplot)

```

```

#Importing data set
getwd()

```

```

## [1] "/home/guest/R/TSA_Sp26t/TSA_Sp26t/Assignments"

```

```

energy_data1 <- read_excel(path=".~/Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Sourc

```

```

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

```

```

read_col_names <- read_excel(path=".~/Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Sou

```

```

## New names:
## * ` ` -> `...1`
## * ` ` -> `...2`
## * ` ` -> `...3`
## * ` ` -> `...4`
## * ` ` -> `...5`
## * ` ` -> `...6`
## * ` ` -> `...7`
## * ` ` -> `...8`

```

```

## * `` -> `...9`  

## * `` -> `...10`  

## * `` -> `...11`  

## * `` -> `...12`  

## * `` -> `...13`  

## * `` -> `...14`  
  

colnames(energy_data1) <- read_col_names  
  

energy_data_processed <- energy_data1 %>%  

  select("Month",  

  "Total Renewable Energy Production",  

  "Hydroelectric Power Consumption") %>%  

  mutate(Month = as.Date(Month, format = "%Y/%m/%d"))  

  summary(energy_data_processed)

```

```

##      Month      Total Renewable Energy Production  

##  Min.   :1973-01-01   Min.   :185.3  

##  1st Qu.:1986-03-01   1st Qu.:311.6  

##  Median :1999-05-01   Median :348.7  

##  Mean   :1999-05-02   Mean   :409.2  

##  3rd Qu.:2012-07-01   3rd Qu.:533.0  

##  Max.   :2025-09-01   Max.   :812.8  

## Hydroelectric Power Consumption  

##  Min.   : 49.02  

##  1st Qu.: 68.89  

##  Median : 78.42  

##  Mean   : 79.36  

##  3rd Qu.: 88.86  

##  Max.   :119.40

```

```

##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.

```

ts_energy_data <- ts(energy_data_processed[,-1], start=c(1973,1), frequency=12)  
  

# Total Renewable Energy Production  

p_ts <- autoplot(ts_energy_data[, "Total Renewable Energy Production"]) +  

  geom_hline(yintercept = mean(ts_energy_data[, "Total Renewable Energy Production"]), na.rm = TRUE), co  

  labs(title = "Total Renewable Energy Production (Monthly)", x = "Year", y = "Energy Production")  

p_acf <- forecast::ggAcf(ts_energy_data[, "Total Renewable Energy Production"], lag.max = 40) + labs(ti  
  

p_pacf <- forecast::ggPacf(ts_energy_data[, "Total Renewable Energy Production"], lag.max = 40) + labs(  
  

cowplot::plot_grid(  

  p_ts, p_acf, p_pacf,  

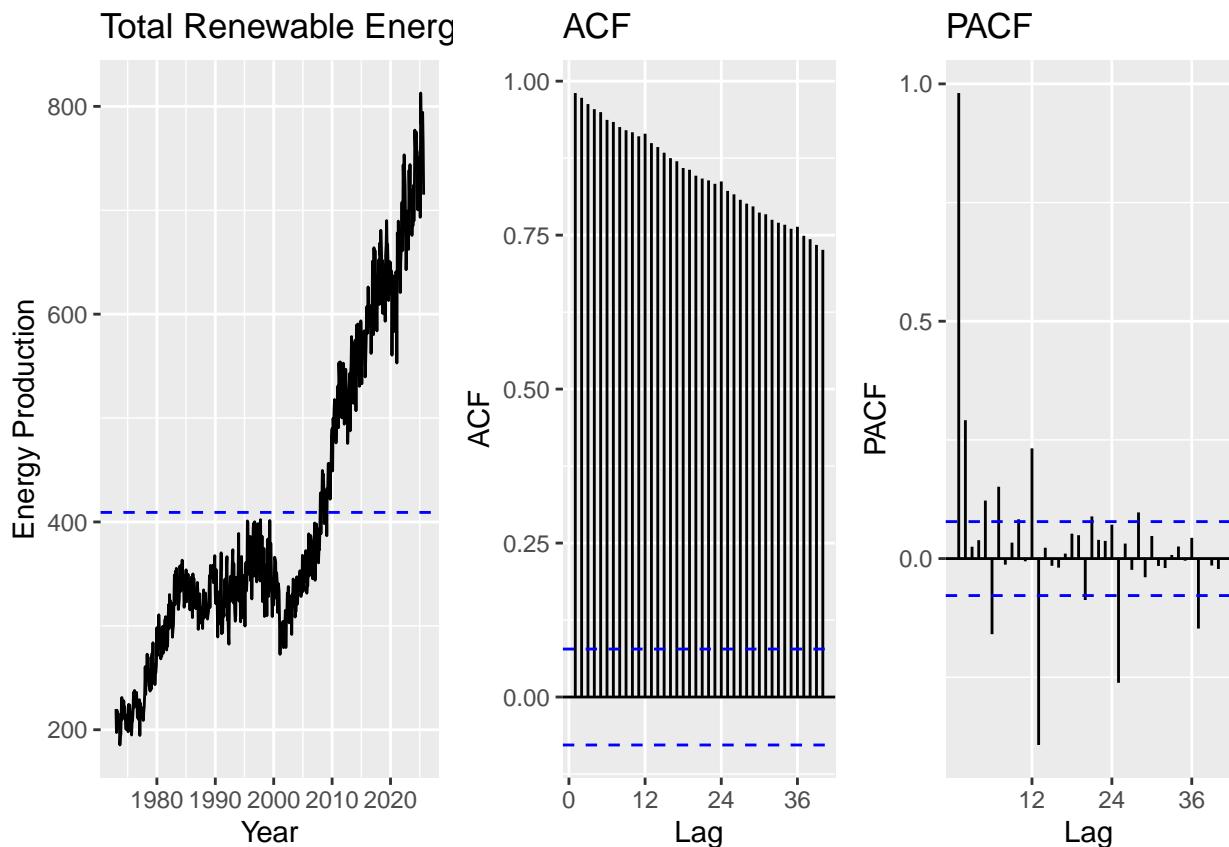
  nrow = 1,

```

```

align = "h",
axis = "tb",
rel_widths = c(2.3, 2, 2))

```



```

#Hydroelectric Power Consumption
p_ts2 <- autoplot(ts_energy_data[, "Hydroelectric Power Consumption"]) +
  geom_hline(yintercept = mean(ts_energy_data[, "Hydroelectric Power Consumption"], na.rm = TRUE), color = "blue")
  labs(title = "Hydroelectric Power Consumption (Monthly)", x = "Year", y = "Energy Consumption")

p_acf2 <- forecast::ggAcf(ts_energy_data[, "Hydroelectric Power Consumption"], lag.max = 40, plot = TRUE)

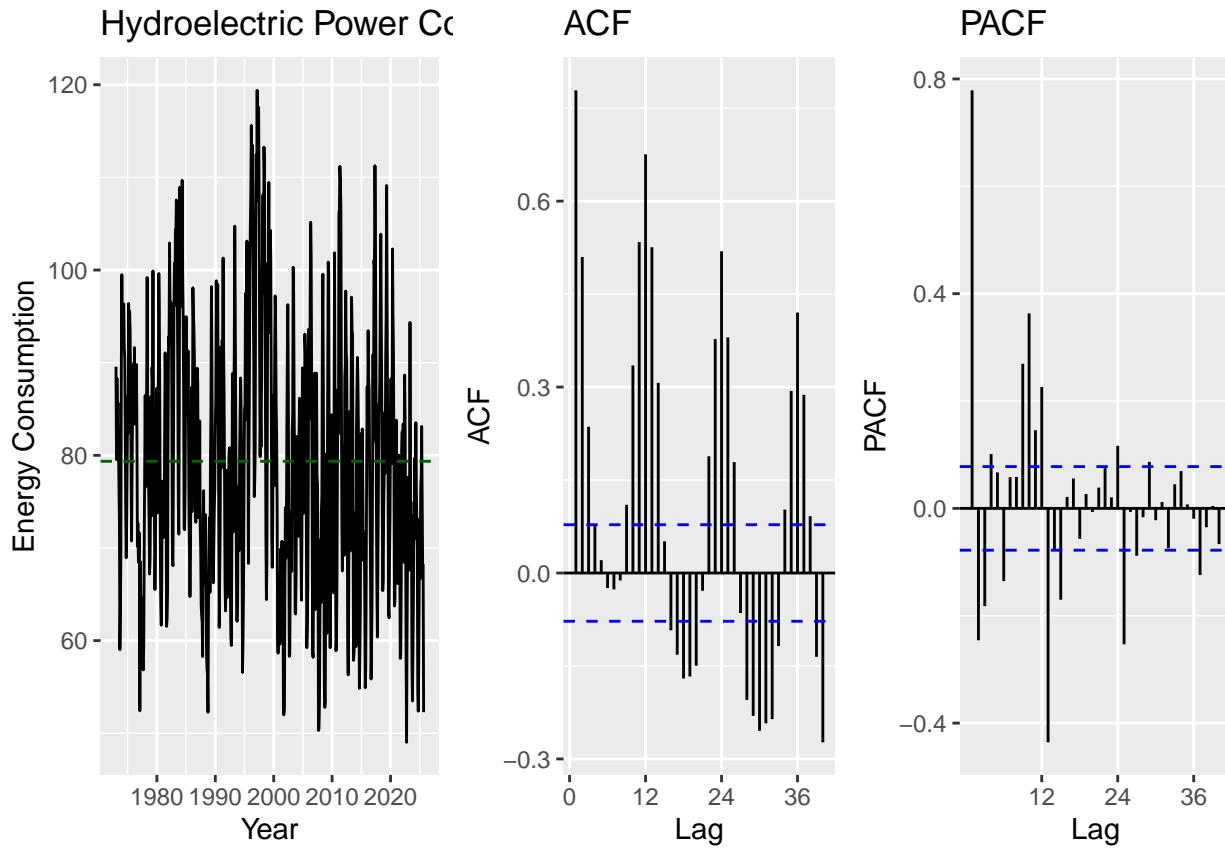
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown
## parameters: `main`

p_pacf2 <- forecast::ggPacf(ts_energy_data[, "Hydroelectric Power Consumption"], lag.max = 40, plot = TRUE)

## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown
## parameters: `main`

cowplot::plot_grid(
  p_ts2, p_acf2, p_pacf2,
  nrow = 1,
  align = "h",
  axis = "tb",
  rel_widths = c(2.3, 2, 2))

```



## 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: From the plots in Q1, Total Renewable Energy Production clearly exhibits a strong upward trend, with accelerated growth in more recent years. What's more, the ACF remains high with very slow decay, which indicates that this serie is non-stationary one. On the other hand, Hydroelectric Power Consumption fluctuates around a relatively stable mean, with a seasonal factor. In addition, the acf and pacf decays quickly from a high value, indicating that the serie is more likely to be a stationary one without a strong long-term trend.

## 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.

```
nobs <- nrow(energy_data_processed)
t <- 1:nobs

lm_renew <- lm(
  ts_energy_data[, "Total Renewable Energy Production"] ~ t)
summary(lm_renew)
```

```
##
```

```

## Call:
## lm(formula = ts_energy_data[, "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

beta0_renew <- as.numeric(lm_renew$coefficients[1]) # intercept
beta1_renew <- as.numeric(lm_renew$coefficients[2]) # slope

lm_hydro <- lm(
  ts_energy_data[, "Hydroelectric Power Consumption"] ~ t)
summary(lm_hydro)

##
## Call:
## lm(formula = ts_energy_data[, "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

beta0_hydro <- as.numeric(lm_hydro$coefficients[1]) # intercept
beta1_hydro <- as.numeric(lm_hydro$coefficients[2]) # slope

```

Total Renewable Energy Production: The slope equals to 0.75, meaning that the total renewable energy production increases on average by about 0.75 units per month. The intercept is 171.45, which means that the estimated level of total renewable energy production when  $t = 0$ . Thus, there is a strong and statistically significant (with  $p\text{-value: } < 2.2\text{e-}16$ ) upward linear trend over time.

Hydroelectric Power Consumption: The

slope equals to  $-0.012$ , so hydroelectric power consumption decreases by about  $0.012$  units per month on average. The intercept is  $83.22$ , hydroelectric power consumption is around  $83.22$  at  $t = 0$ . This series has a very slight downward linear trend, but the magnitude of the trend is negligible, and the series largely fluctuates around a stable mean.

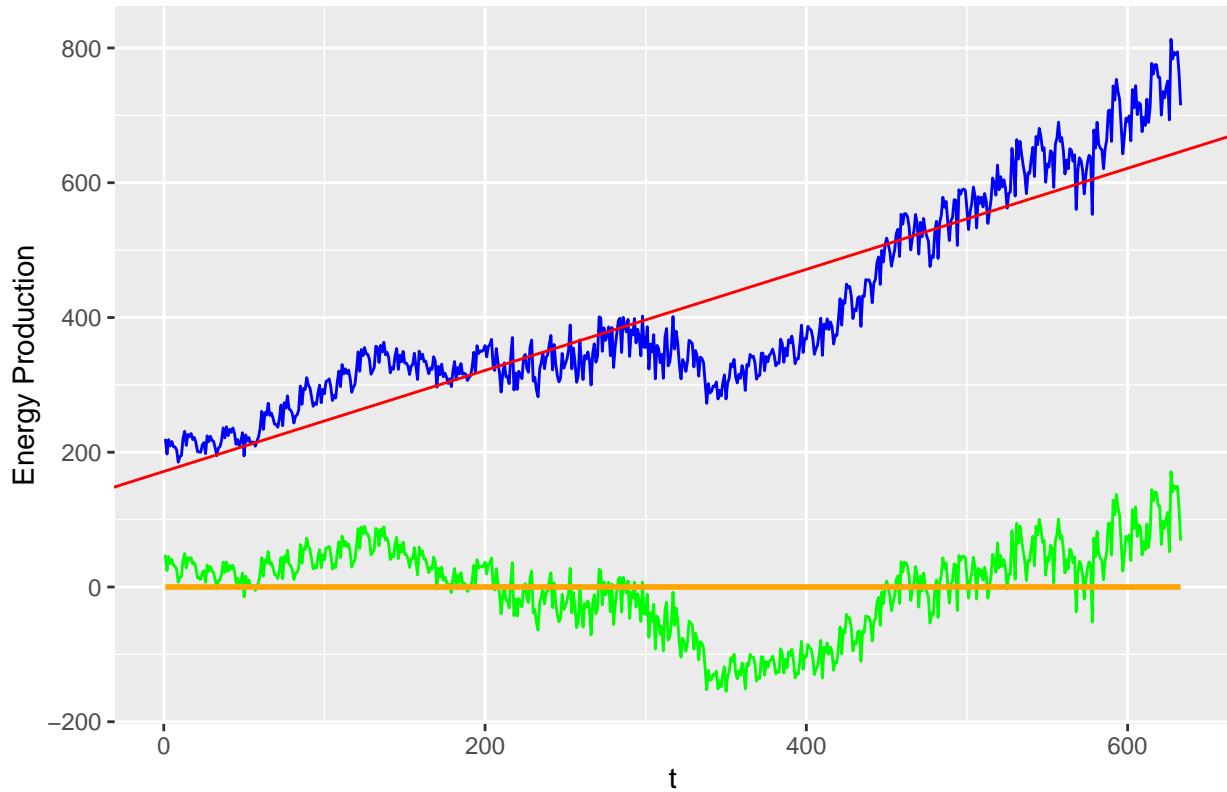
#### Q4

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.

```
# detrend using regression coefficients
detrend_renew <- ts_energy_data[, "Total Renewable Energy Production"] - (beta0_renew + beta1_renew * t)
detrend_hydro <- ts_energy_data[, "Hydroelectric Power Consumption"] - (beta0_hydro + beta1_hydro * t)
ts_detrend_renew <- ts(detrend_renew,
                         frequency = 12,
                         start = c(1973, 1))
ts_detrend_hydro <- ts(detrend_hydro,
                         frequency = 12,
                         start = c(1973, 1))
ggplot(energy_data_processed,
       aes(x = t, y = `Total Renewable Energy Production`)) +
  geom_line(color = "blue") +
  geom_abline(intercept = beta0_renew, slope = beta1_renew, color = "red") +
  geom_line(aes(y = detrend_renew), color = "green") +
  geom_smooth(aes(y = detrend_renew), method = "lm",
              color = "orange", se = FALSE) +
  labs(title = "Total Renewable Energy Production: Original vs Detrended", y = "Energy Production")

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

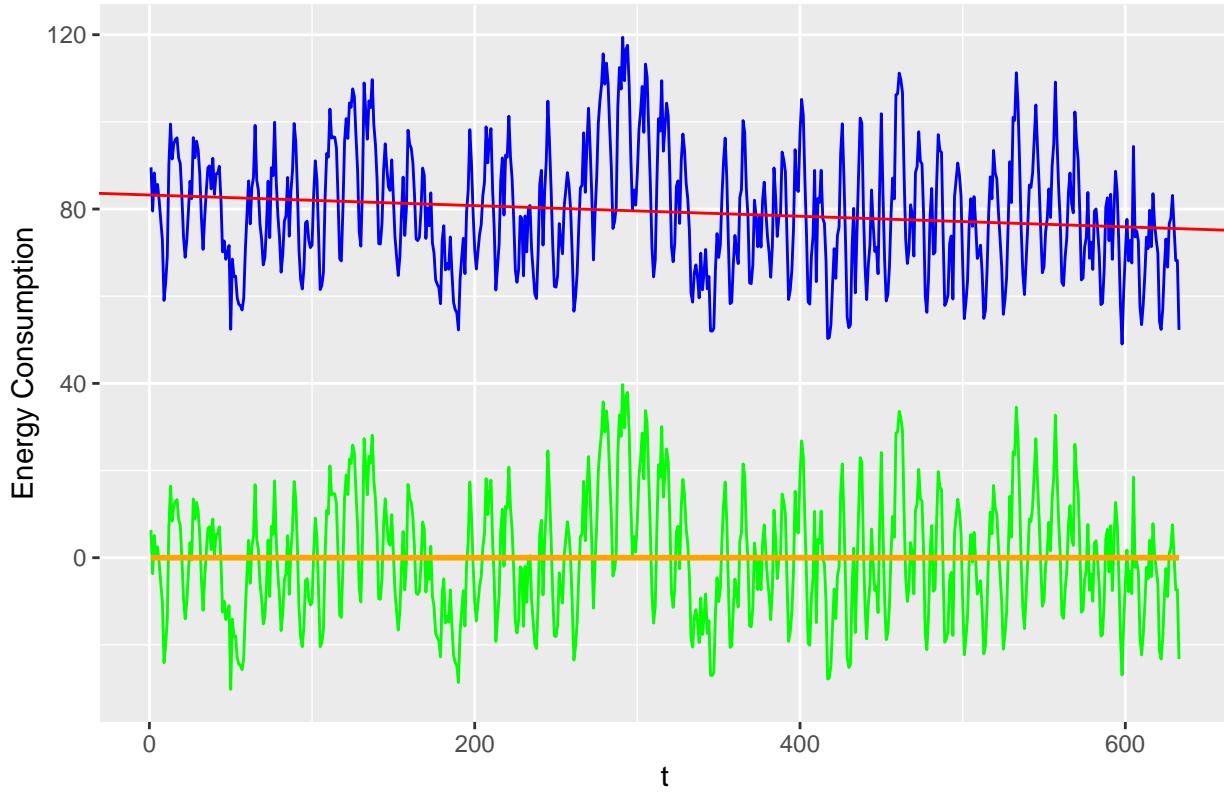
## Total Renewable Energy Production: Original vs Detrended



```
ggplot(energy_data_processed,
       aes(x = t, y = `Hydroelectric Power Consumption`)) +
  geom_line(color = "blue") +
  geom_abline(intercept = beta0_hydro, slope = beta1_hydro, color = "red") +
  geom_line(aes(y = detrend_hydro), color = "green") +
  geom_smooth(aes(y = detrend_hydro), method = "lm",
              color = "orange", se = FALSE) +
  labs(title = "Hydroelectric Power Consumption: Original vs Detrended", y = "Energy Consumption")

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

## Hydroelectric Power Consumption: Original vs Detrended



Total Renewable Energy Production: The original series shows a strong, persistent upward trend over time, but detrending successfully removes this strong upward trend. Thus, the a series that is approximately stationary in mean, showing a fluctuation around 0. Hydroelectric Power Consumption: The original series fluctuates around a relatively constant level with no obvious long-term trend, and the detrended series still looks very similar to the original. Thus, detrending has minimal impact on hydroelectric power consumption, confirming that this series does not exhibit a strong linear trend.

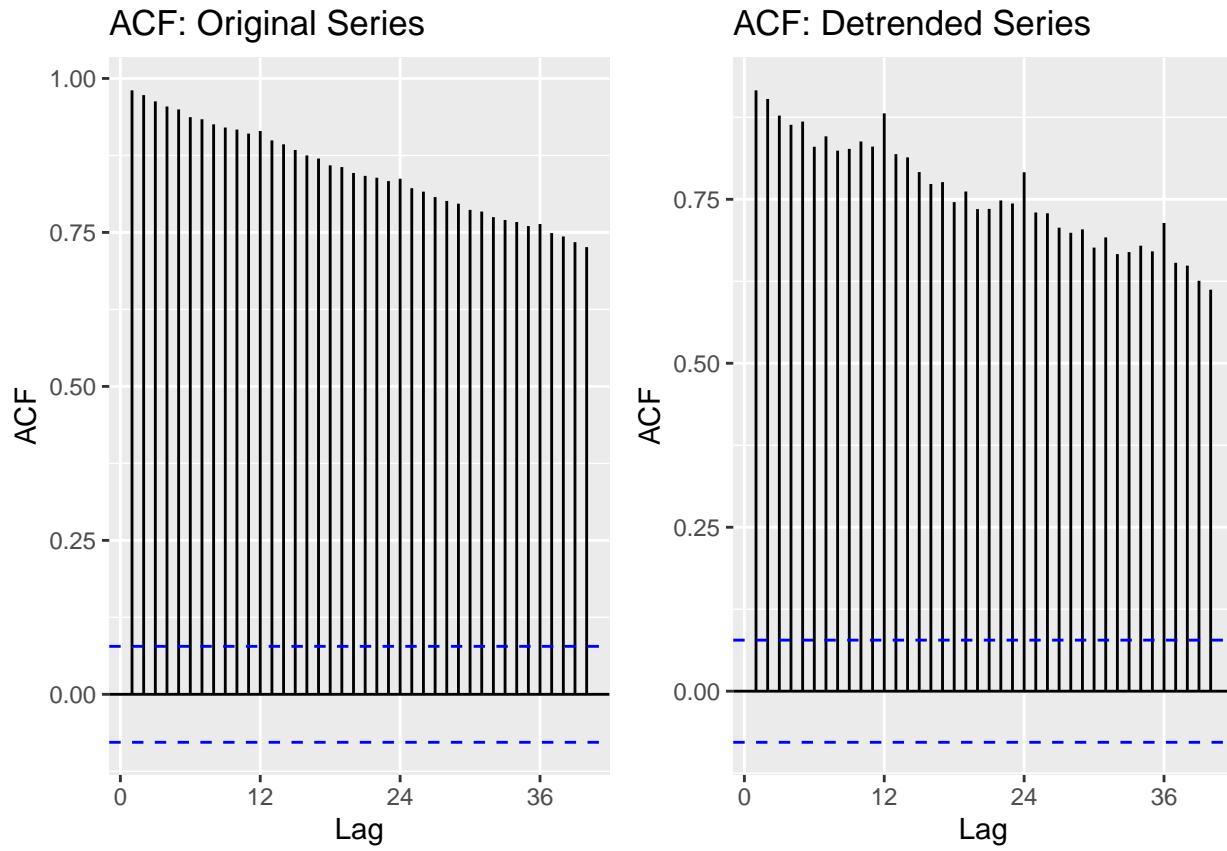
### Q5

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?

```
# Renew - ACF
p_acf_orig_renew <- ggAcf(ts_energy_data[, "Total Renewable Energy Production"],
  lag.max = 40) + labs(title = "ACF: Original Series")

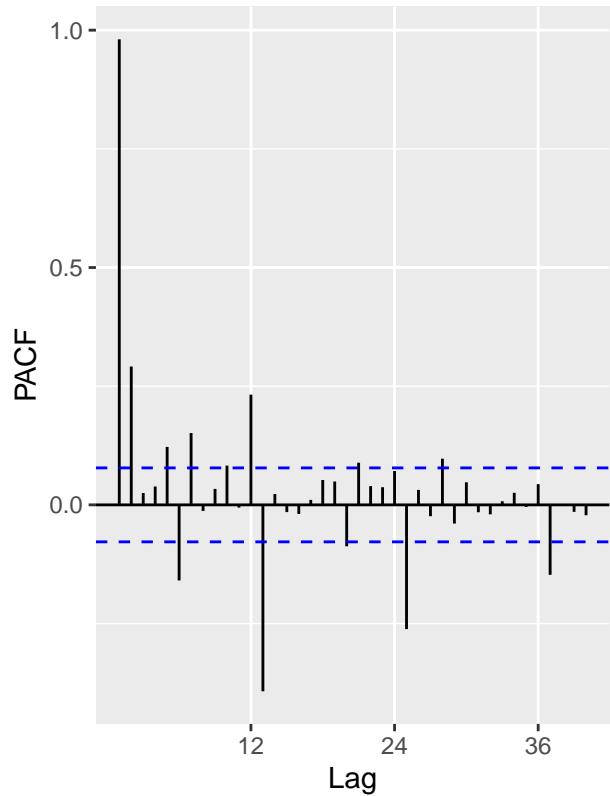
p_acf_det_renew <- ggAcf(ts_detrend_renew, lag.max = 40) + labs(title = "ACF: Detrended Series")

cowplot::plot_grid(p_acf_orig_renew, p_acf_det_renew, nrow = 1)
```

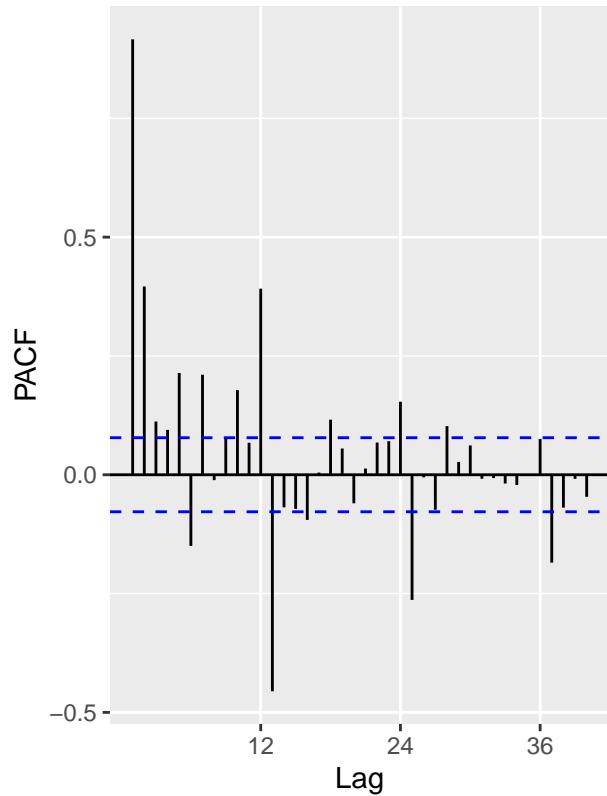


```
# Renew - PACF
p_pacf_orig_renew <- ggPacf(ts_energy_data[, "Total Renewable Energy Production"], lag.max = 40) + labs(
  title = "ACF: Original Series")
p_pacf_det_renew <- ggPacf(ts_detrend_renew, lag.max = 40) + labs(title = "ACF: Detrended Series")
cowplot::plot_grid(p_pacf_orig_renew, p_pacf_det_renew, nrow = 1)
```

PACF: Original Series

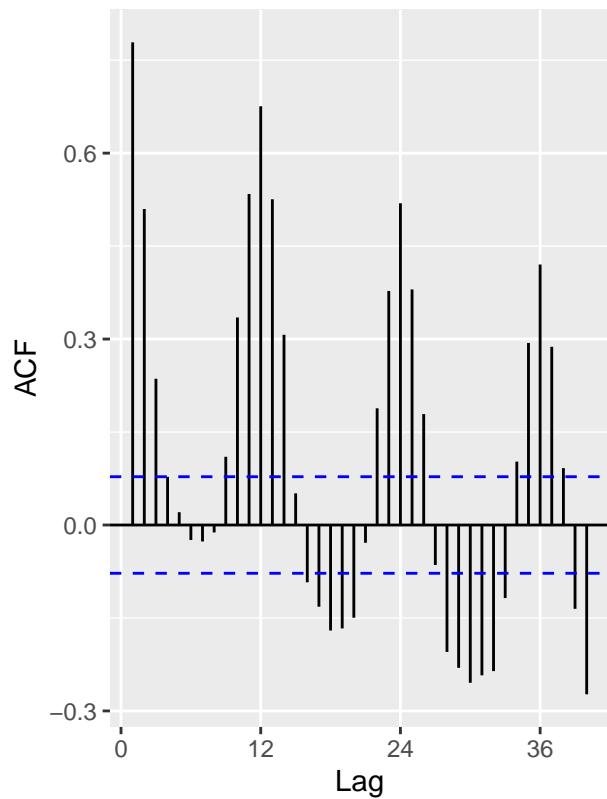


PACF: Detrended Series

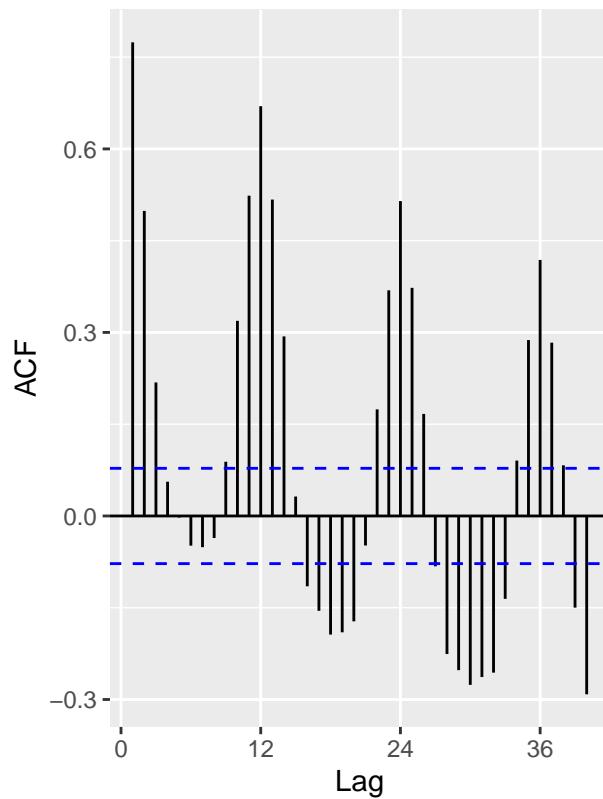


```
# Hydro - ACF
p_acf_orig_hydro <- ggAcf( ts_energy_data[, "Hydroelectric Power Consumption"], lag.max = 40) + labs(title = "PACF: Original Series")
p_acf_det_hydro <- ggAcf(ts_detrend_hydro, lag.max = 40) + labs(title = "PACF: Detrended Series")
cowplot::plot_grid(p_acf_orig_hydro, p_acf_det_hydro, nrow = 1)
```

ACF: Original Series



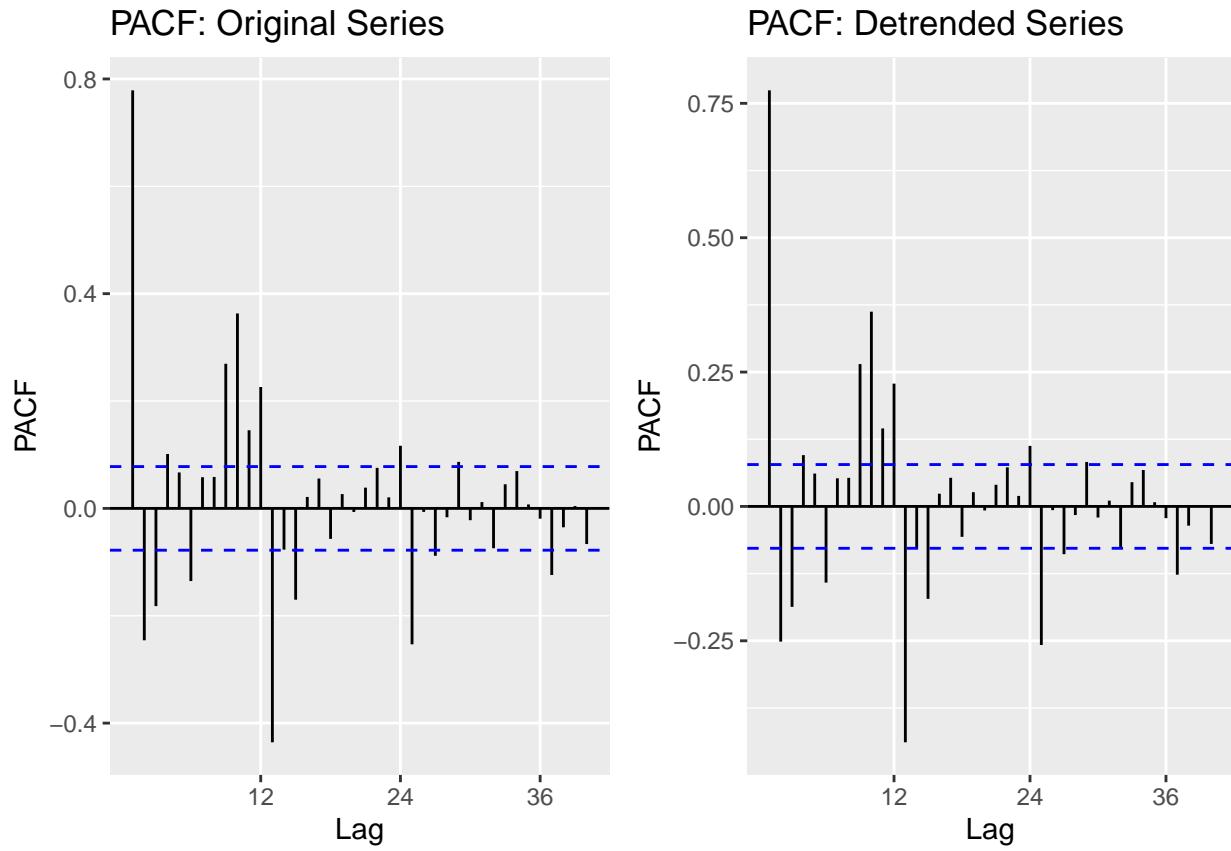
ACF: Detrended Series



```
# Hydro - PACF
p_pacf_orig_hydro <- ggPacf(ts_energy_data[, "Hydroelectric Power Consumption"], lag.max = 40) + labs(title = "PACF: Original Series")

p_pacf_det_hydro <- ggPacf(ts_detrend_hydro, lag.max = 40) + labs(title = "PACF: Detrended Series")

cowplot::plot_grid(p_pacf_orig_hydro, p_pacf_det_hydro, nrow = 1)
```



After detrending, the ACF of the renewable energy series shows reduced autocorrelation and a slower decay. So, part of the non-stationarity in the original series was coming from a deterministic trend. The PACF also shows fewer large spikes and a clearer seasonal effect, but the significant autocorrelation remains, suggesting that seasonality and short-term dependence still exists.

For hydroelectric power consumption, detrending does not change the ACF or PACF by a great amount. The strong seasonal trend remains dominant, with a significant short-term change for both. The series is primarily driven by seasonality rather than a deterministic 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 shows a dominant long-term upward trend, without any obvious fluctuations. In the ACF, there are small bumps at seasonal lags, but they are relatively weak compared to the overall persistence caused by the trend, suggesting that seasonality is present but not a dominant feature of the series. On the other hand for Hydroelectric Power Consumption, the time series displays a repeating within-year oscillations. In addition, the ACF exhibits some strong spikes at seasonal lags, showing a seasonal behavior. There is a strong annual seasonal pattern in hydroelectric power consumption.

## 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 your answer to Q6?

```
dummies_renew <- seasonaldummy(ts_detrend_renew)

seas_means_renew <- lm(detrend_renew ~ dummies_renew)
summary(seas_means_renew)

## 
## Call:
## lm(formula = detrend_renew ~ dummies_renew)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -153.09  -36.94   15.01   42.21  155.62 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)    7.320     8.763   0.835   0.4039    
## dummies_renewJan  5.840    12.334   0.473   0.6360    
## dummies_renewFeb -31.525    12.334  -2.556   0.0108 *  
## dummies_renewMar  8.205    12.334   0.665   0.5061    
## dummies_renewApr -5.400    12.334  -0.438   0.6617    
## dummies_renewMay  8.912    12.334   0.723   0.4703    
## dummies_renewJun -2.231    12.334  -0.181   0.8565    
## dummies_renewJul  3.114    12.334   0.252   0.8008    
## dummies_renewAug -5.478    12.334  -0.444   0.6571    
## dummies_renewSep -31.283    12.334  -2.536   0.0114 *  
## dummies_renewOct -18.437    12.393  -1.488   0.1373    
## dummies_renewNov -19.867    12.393  -1.603   0.1094    
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 63.19 on 621 degrees of freedom
## Multiple R-squared:  0.04701,   Adjusted R-squared:  0.03013 
## F-statistic: 2.785 on 11 and 621 DF,  p-value: 0.00152

beta0_renew_seas <- seas_means_renew$coefficients[1]
beta_renew_seas <- seas_means_renew$coefficients[2:12]

dummies_hydro <- seasonaldummy(ts_detrend_hydro)
seas_means_hydro <- lm(detrend_hydro ~ dummies_hydro)
summary(seas_means_hydro)

## 
## Call:
## lm(formula = detrend_hydro ~ dummies_hydro)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -153.09  -36.94   15.01   42.21  155.62 
```

```

## -34.116 -5.871 -0.555 5.823 32.264
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.3796    1.4030   0.271 0.786811
## dummies_hydroJan   4.8900    1.9747   2.476 0.013541 *
## dummies_hydroFeb  -2.4641    1.9747  -1.248 0.212573
## dummies_hydroMar   7.0794    1.9747   3.585 0.000364 ***
## dummies_hydroApr   5.5895    1.9747   2.830 0.004798 **
## dummies_hydroMay  14.0676    1.9747   7.124 2.92e-12 ***
## dummies_hydroJun  10.7799    1.9747   5.459 6.93e-08 ***
## dummies_hydroJul   4.0156    1.9747   2.033 0.042427 *
## dummies_hydroAug  -5.2952    1.9747  -2.681 0.007525 **
## dummies_hydroSep -16.5612    1.9747  -8.386 3.37e-16 ***
## dummies_hydroOct -16.3534    1.9841  -8.242 1.01e-15 ***
## dummies_hydroNov -10.7940    1.9841  -5.440 7.67e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.12 on 621 degrees of freedom
## Multiple R-squared:  0.4827, Adjusted R-squared:  0.4735
## F-statistic: 52.67 on 11 and 621 DF,  p-value: < 2.2e-16

beta0_hydro_seas <- seas_means_hydro$coefficients[1]
beta_hydro_seas <- seas_means_hydro$coefficients[2:12]

```

The results all matches my answer for Q6. For total renewable energy production, only two of the monthly dummy coefficients are statistically significant. The r-square is less than 5%, indicates that this series does not exhibit a seasonal trend. Then, for hydroelectric power consumption, almost all monthly dummy coefficients are statistically significant, and the values of the dummy coefficients shows that there are usually high consumption in spring/early summer, and relative low in late summer/fall. The overall p-value is  $< 2.2e-16$ , showing that this series has a strong seasonal trend.

## 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?

```

nobs <- length(detrend_renew)
renew_seas_comp <- numeric(nobs)

for(i in 1:nobs){
  renew_seas_comp[i] <- beta0_renew_seas + beta_renew_seas %*% dummies_renew[i,]
}
hydro_seas_comp <- numeric(nobs)

for(i in 1:nobs){
  hydro_seas_comp[i] <- beta0_hydro_seas + beta_hydro_seas %*% dummies_hydro[i,]
}

# deseason detrended renewable series
deseason_renew <- detrend_renew - renew_seas_comp

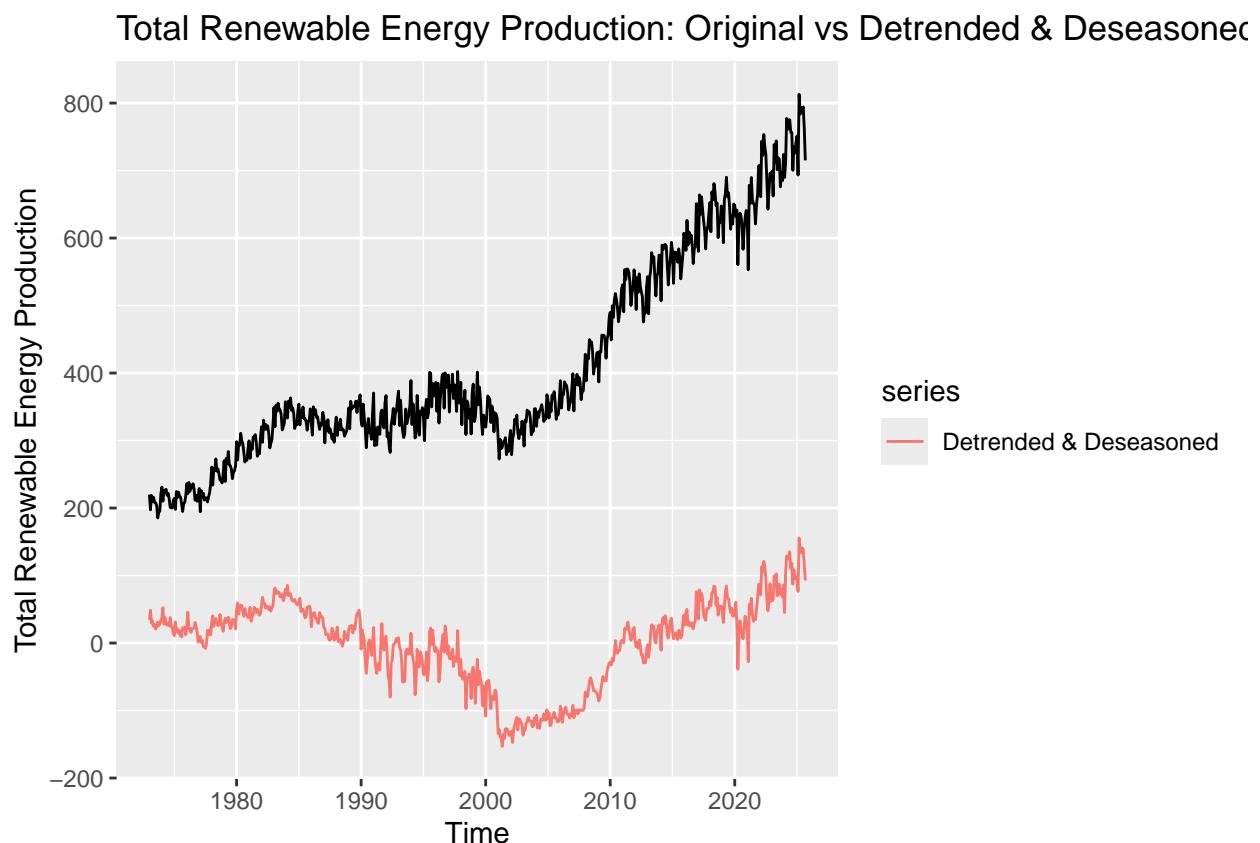
```

```

# convert to ts object
ts_deseason_renew <- ts(deseason_renew, frequency = 12, start = c(1973, 1))

autoplot(ts_energy_data[, "Total Renewable Energy Production"]) +
  autolayer(ts_deseason_renew,
            series = "Detrended & Deseasoned") +
  ylab("Total Renewable Energy Production") +
  labs(title = "Total Renewable Energy Production: Original vs Detrended & Deseasoned")

```



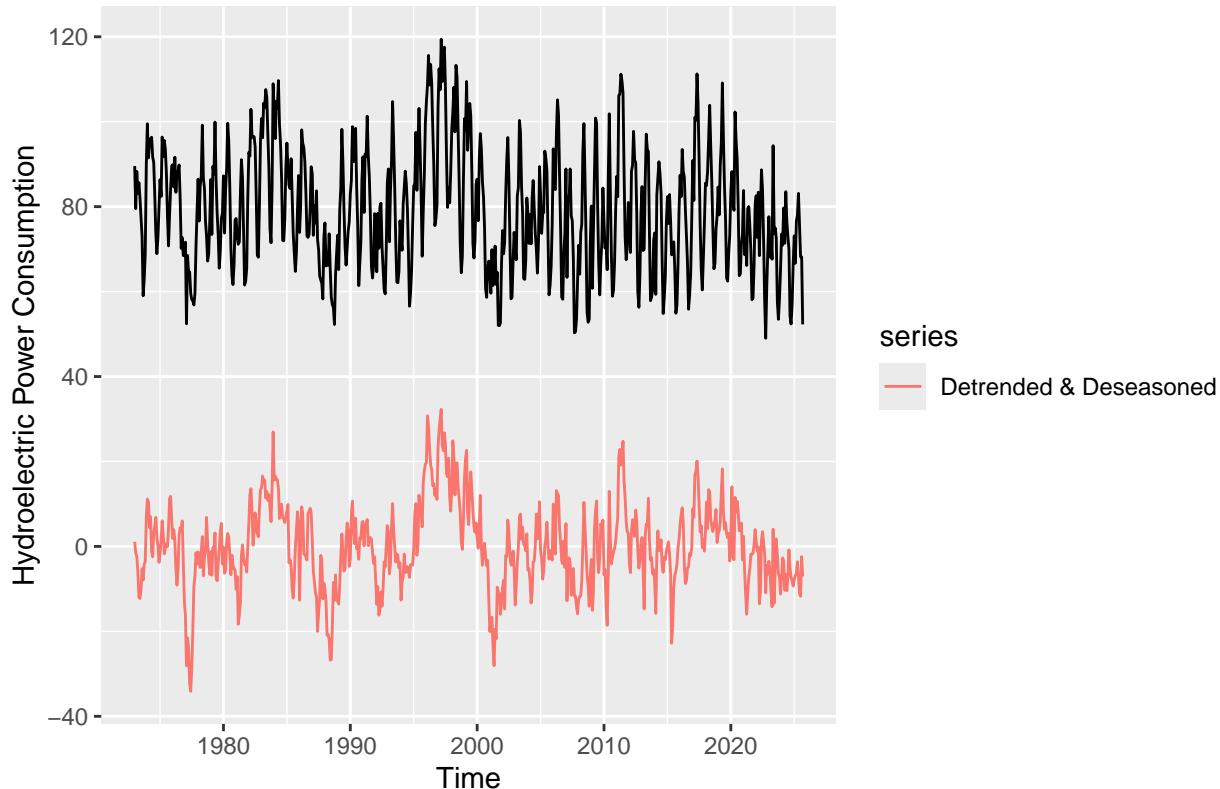
```

# deseason detrended hydro series
deseason_hydro <- detrend_hydro - hydro_seas_comp
# convert to ts object
ts_deseason_hydro <- ts(deseason_hydro, frequency = 12, start = c(1973, 1))

autoplot(ts_energy_data[, "Hydroelectric Power Consumption"]) +
  autolayer(ts_deseason_hydro,
            series = "Detrended & Deseasoned") +
  ylab("Hydroelectric Power Consumption") +
  labs(title = "Hydroelectric Power Consumption: Original vs Detrended & Deseasoned")

```

## Hydroelectric Power Consumption: Original vs Detrended & Deseasoned



For total renewable energy production, deseasoning alone does not substantially change the behavior of the series compared to Q1. So, that trend instead of seasonality is the dominant source of non-stationarity for this series.

Deseasoning produces a substantial change for hydroelectric power consumption, and the time plot becomes much flatter. The strong cycle showed in Q1 disappears, and the ACF and PACF no longer contains dominant seasonal lags. It confirms that that seasonality was the main driver of its dependence structure.

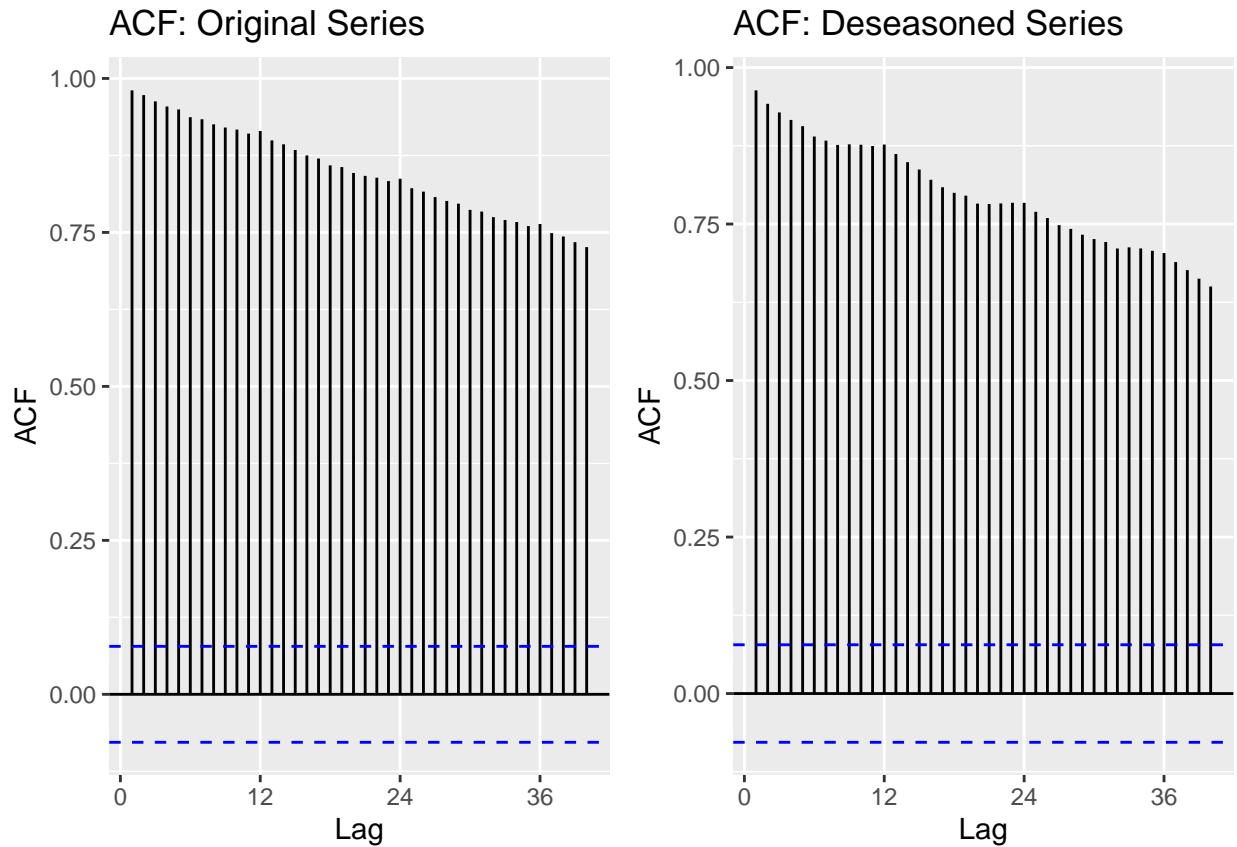
### Q9

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?

```
# Renew - ACF
p_acf_orig_renew <- ggAcf(ts_energy_data[, "Total Renewable Energy Production"], lag.max = 40) + labs(title = "ACF: Original Series")

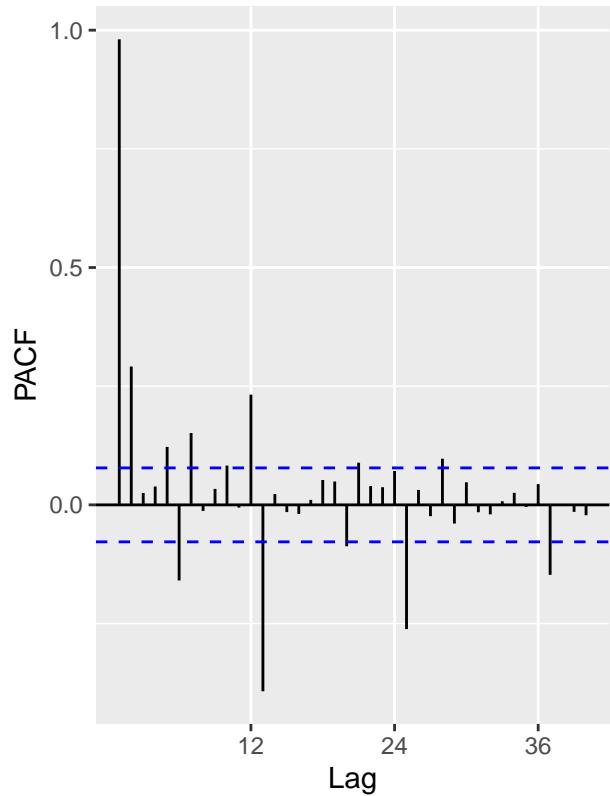
p_acf_deseason_renew <- ggAcf(ts_deseason_renew, lag.max = 40) + labs(title = "ACF: Deseasoned Series")

cowplot::plot_grid(p_acf_orig_renew, p_acf_deseason_renew, nrow = 1)
```

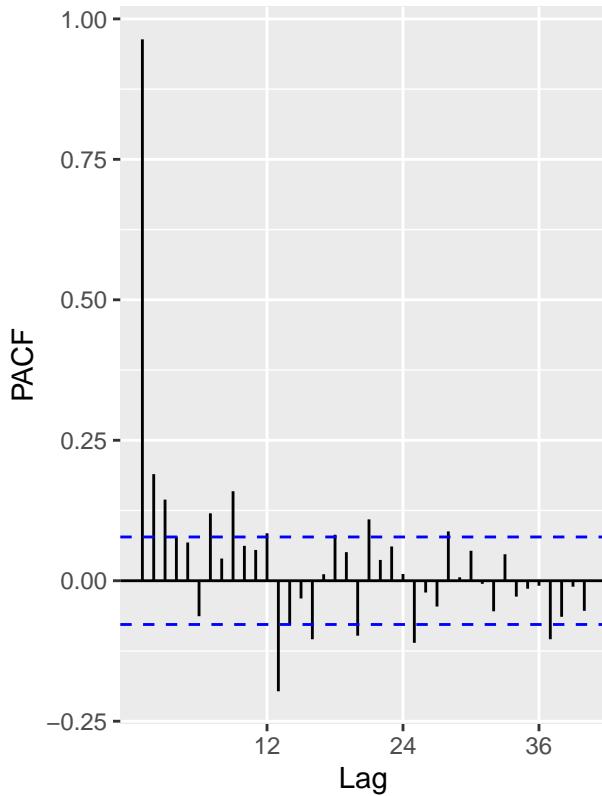


```
# Renew - PACF
p_pacf_orig_renew <- ggPacf(ts_energy_data[, "Total Renewable Energy Production"], lag.max = 40) + labs(
  title = "PACF: Original Series")
p_pacf_deseason_renew <- ggPacf( ts_deseason_renew, lag.max = 40) + labs(title = "PACF: Deseasoned Series")
cowplot::plot_grid(p_pacf_orig_renew, p_pacf_deseason_renew, nrow = 1)
```

PACF: Original Series

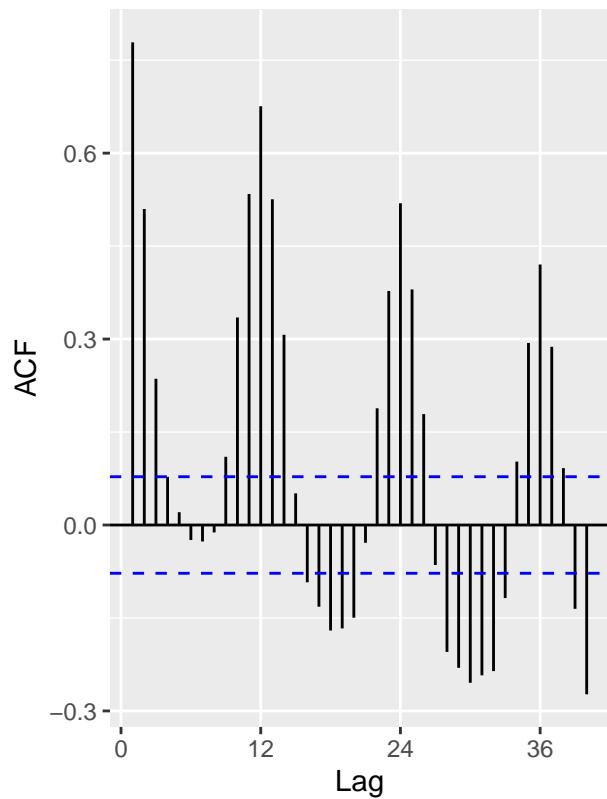


PACF: Deseasoned Series

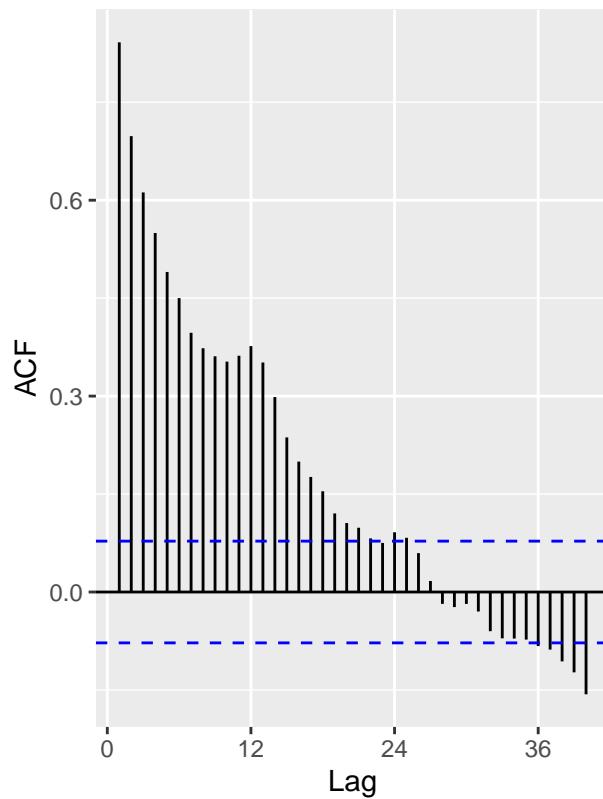


```
# Hydro - ACF
p_acf_orig_hydro <- ggAcf(ts_energy_data[, "Hydroelectric Power Consumption"], lag.max = 40) + labs(title = "PACF: Original Series")
p_acf_deseason_hydro <- ggAcf(ts_deseason_hydro, lag.max = 40) + labs(title = "PACF: Deseasoned Series")
cowplot:::plot_grid(p_acf_orig_hydro, p_acf_deseason_hydro, nrow = 1)
```

ACF: Original Series

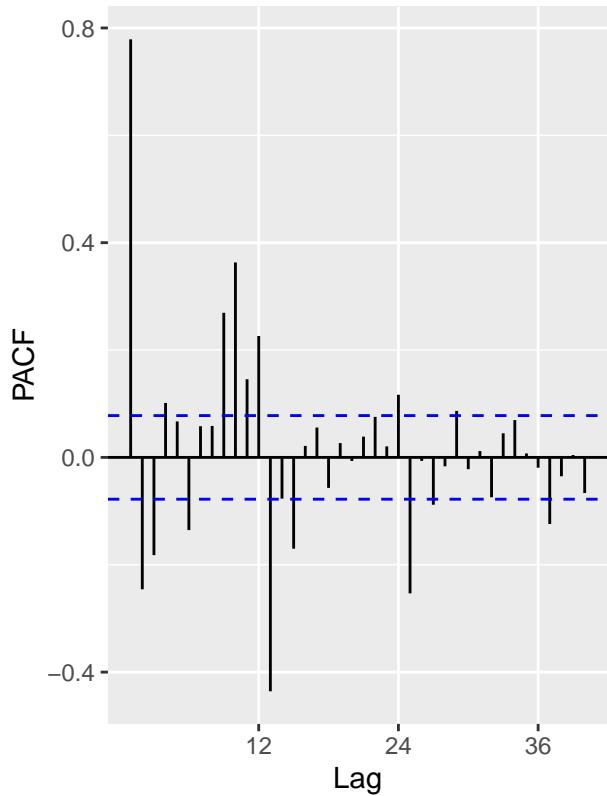


ACF: Deseasoned Series

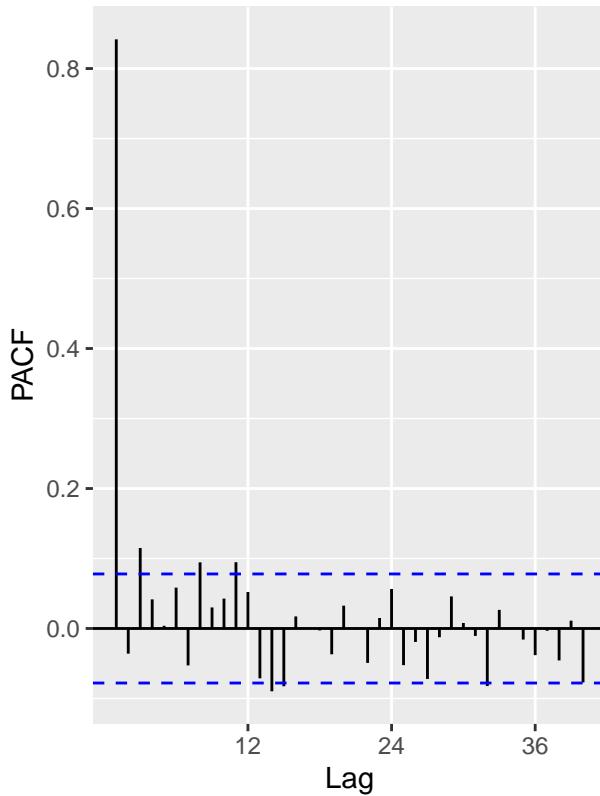


```
# Hydro - ACF
p_pacf_orig_hydro <- ggPacf(ts_energy_data[, "Hydroelectric Power Consumption"], lag.max = 40) + labs(title = "ACF: Original Series")
p_pacf_deseason_hydro <- ggPacf(ts_deseason_hydro, lag.max = 40) + labs(title = "PACF: Deseasoned Series")
cowplot::plot_grid(p_pacf_orig_hydro, p_pacf_deseason_hydro, nrow = 1)
```

PACF: Original Series



PACF: Deseasoned Series



Compared to the original plots in Q1, deseasoning has little effect on the ACF and PACF of total renewable energy production, as the series remains dominated by a strong trend and slow autocorrelation decay.

In contrast, deseasoning substantially changes the autocorrelation structure of hydroelectric power consumption by removing some of the dominant seasonal spikes. As a result, the deseasoned hydro series exhibits faster decay and fewer significant lags, indicating that seasonality was the primary source of dependence in the original data.