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

Assignment 3 - Due date 02/03/26

Yeeun Kim

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

```r
#Importing data
energy_data1 <- read_excel(path="/Users/yeeunkim/Library/CloudStorage/OneDrive-DukeUniversity/2026 Spri

# Extract the column names from row 11
read_col_names <- read_excel(path="/Users/yeeunkim/Library/CloudStorage/OneDrive-DukeUniversity/2026 Spr

#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 the columns
energy_select <- energy_data1 %>%
  select(1,5:6)
head(energy_select)
```

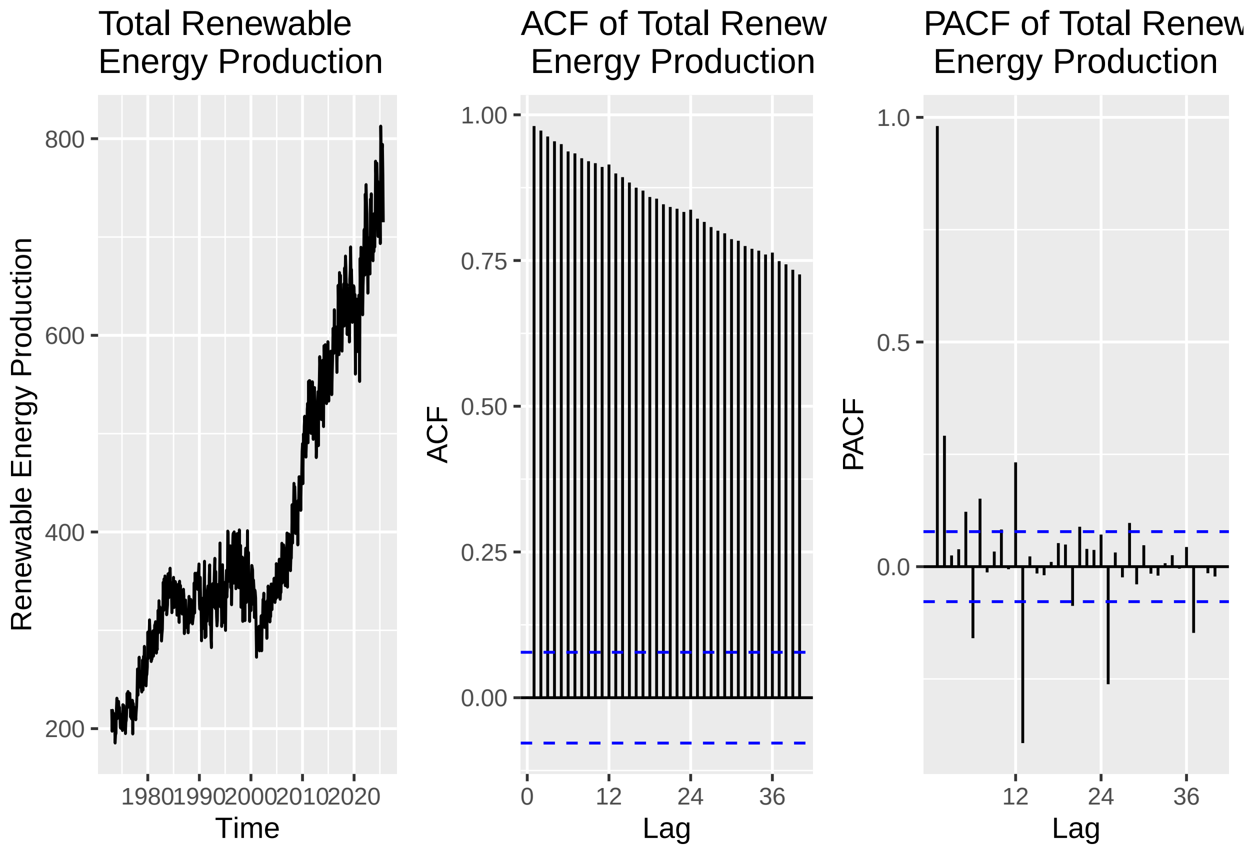
```
## # A tibble: 6 x 3
##   Month               `Total Renewable Energy Production` Hydroelectric Power ~1
##   <dttm>                                            <dbl>                  <dbl>
## 1 1973-01-01 00:00:00                                220.                   89.6
## 2 1973-02-01 00:00:00                                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

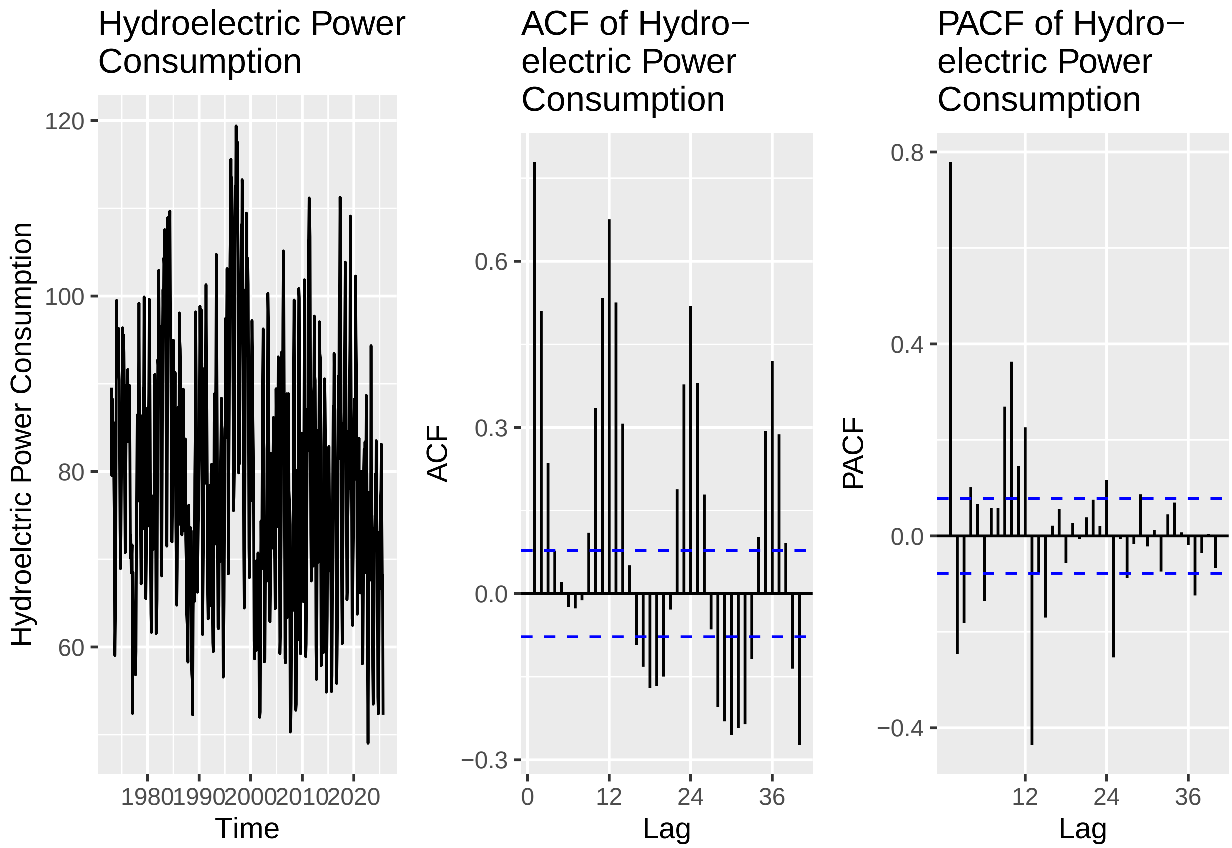
Q1

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.

```r
#TS data
ts_energy_data <- ts(energy_select[,2:3],start = c(1973,1), frequency = 12)

#Renewable Energy Plots
renew_ts_plot <- autoplot(ts_energy_data[,1]) +
  labs(
    title = "Total Renewable\nEnergy Production",
    x = "Time",
    y = "Renewable Energy Production",
    color = NULL)
renew_acf_plot = ggAcf(ts_energy_data[,1], lag.max = 40) +ggtitle("ACF of Total Renewable\n Energy Produ
renew_pacf_plot = ggPacf(ts_energy_data[,1], lag.max = 40) +ggtitle("PACF of Total Renewable\n Energy Pr

all_plot_renew <- plot_grid(renew_ts_plot, renew_acf_plot, renew_pacf_plot, nrow=1)

#Hydroelectric Power Plots
hydro_ts_plot <- autoplot(ts_energy_data[,2]) +
  labs(
    title = "Hydroelectric Power\nConsumption",
    x = "Time",
    y = "Hydroelctric Power Consumption",
    color = NULL)
hydro_acf_plot = ggAcf(ts_energy_data[,2], lag.max = 40) +ggtitle("ACF of Hydro-\nelectric Power\nConsum
hydro_pacf_plot = ggPacf(ts_energy_data[,2], lag.max = 40) +ggtitle("PACF of Hydro-\nelectric Power\nCon

all_plot_hydro <- plot_grid(hydro_ts_plot, hydro_acf_plot, hydro_pacf_plot, nrow=1)

#Results
all_plot_renew
```




```
all_plot_hydro
```



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?

The Total Renewable Energy Production shows an upward trend over time. It soars up in 2010, and ACF decays slowly, indicating a non-stationary series with strong trend.
The Hydroelectric Power Consumption does not show a trend, but shows seasonality as evidenced by the ACF and PACF plots.

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 vector t
t <- 1:length(ts_energy_data[,1])

#Renewable_Linear model
renew_linear_model <- lm(ts_energy_data[,1]~t)
summary(renew_linear_model)
```

```
##
```

```
## Call:
## lm(formula = ts_energy_data[, 1] ~ 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
#Renewable_Store regression coefficients
beta0=as.numeric(renew_linear_model$coefficients[1])  #first coefficient is the intercept term or beta0
beta1=as.numeric(renew_linear_model$coefficients[2])  #second coefficient is the slope or beta1

#Hydro_Linear model
hydro_linear_model <- lm(ts_energy_data[,2]~t)
summary(hydro_linear_model)
```

```
##
## Call:
## lm(formula = ts_energy_data[, 2] ~ 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
#Hydro_Store regression coefficients
beta00=as.numeric(hydro_linear_model$coefficients[1])  #first coefficient is the intercept term or beta
beta11=as.numeric(hydro_linear_model$coefficients[2])  #second coefficient is the slope or beta1
```

For total renewable energy production, the slope is 0.74999, which is positive and statistically significant (p-value $<$ 2.2e-16), meaning an upward trend in renewable energy production over time, with production increasing by around 0.75 units per month on average. The intercept is 171.45. The adjusted R-squared value is 0.8201, indicating that the linear time trend explains 82% of the variation in renewable energy production.

However, hydroelectric power consumption has an slope of -0.012, which is negative and statistically significant (p-value = 6.547e-05), suggesting a downward trend over time. The intercept is 83.223802. The adjusted R-squared is 0.02342, meaning that the linear trend explains 2% of the total variation in the series. This low adjusted R-squared indicates that time is not an important explanatory factor for hydroelectric power consumption.

Q4

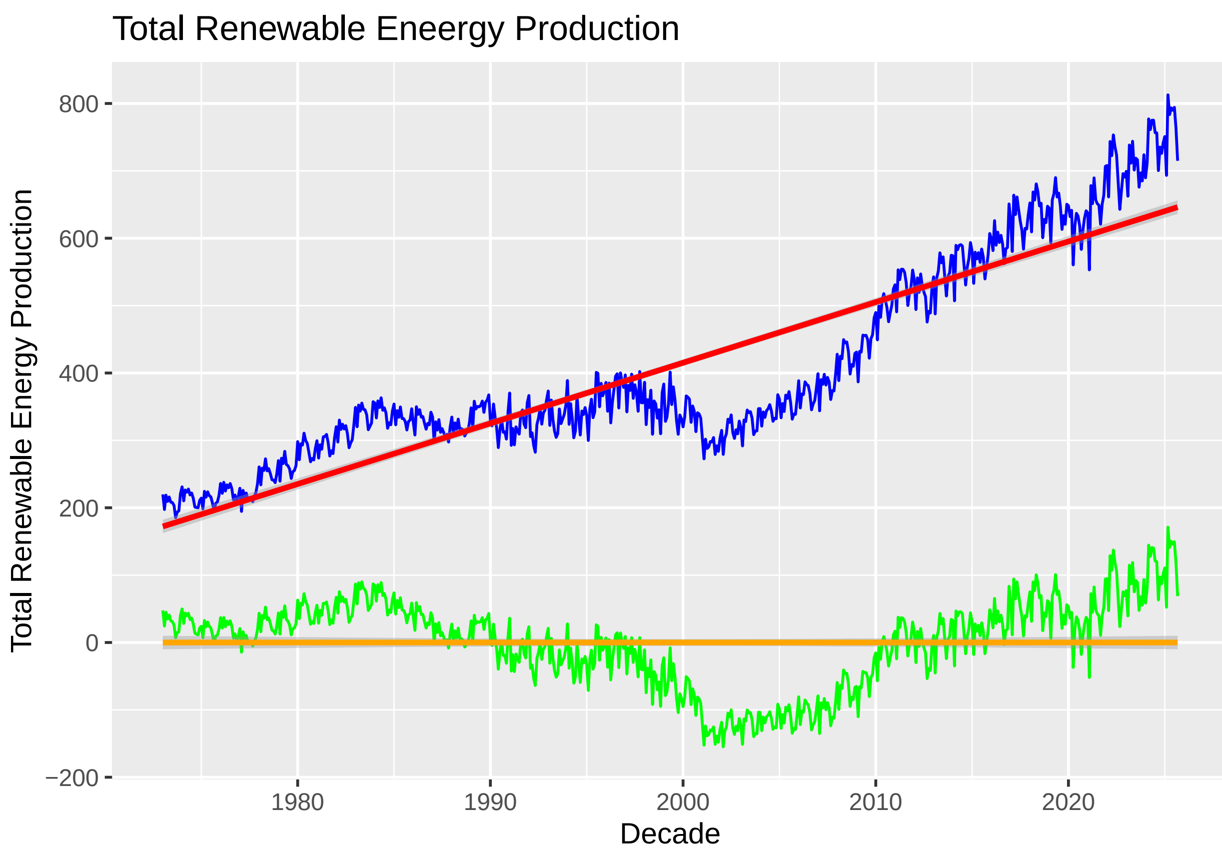
Use the regression coefficients to detrend each series (subtract fitted linear trend). Plot detrended series and compare with the original time series from Q1. Describe what changed.

Detrending removes the upward trend in renewable energy production, but leaves the hydroelectric power consumption series unchanged.

```r
#Renewable: remove the trend from series
renew_detrend_data <- energy_select[,2]-(beta0+beta1*t)

#Renewable: note detrend will be a data frame and not a ts object
class(renew_detrend_data)
```

```
## [1] "data.frame"
```

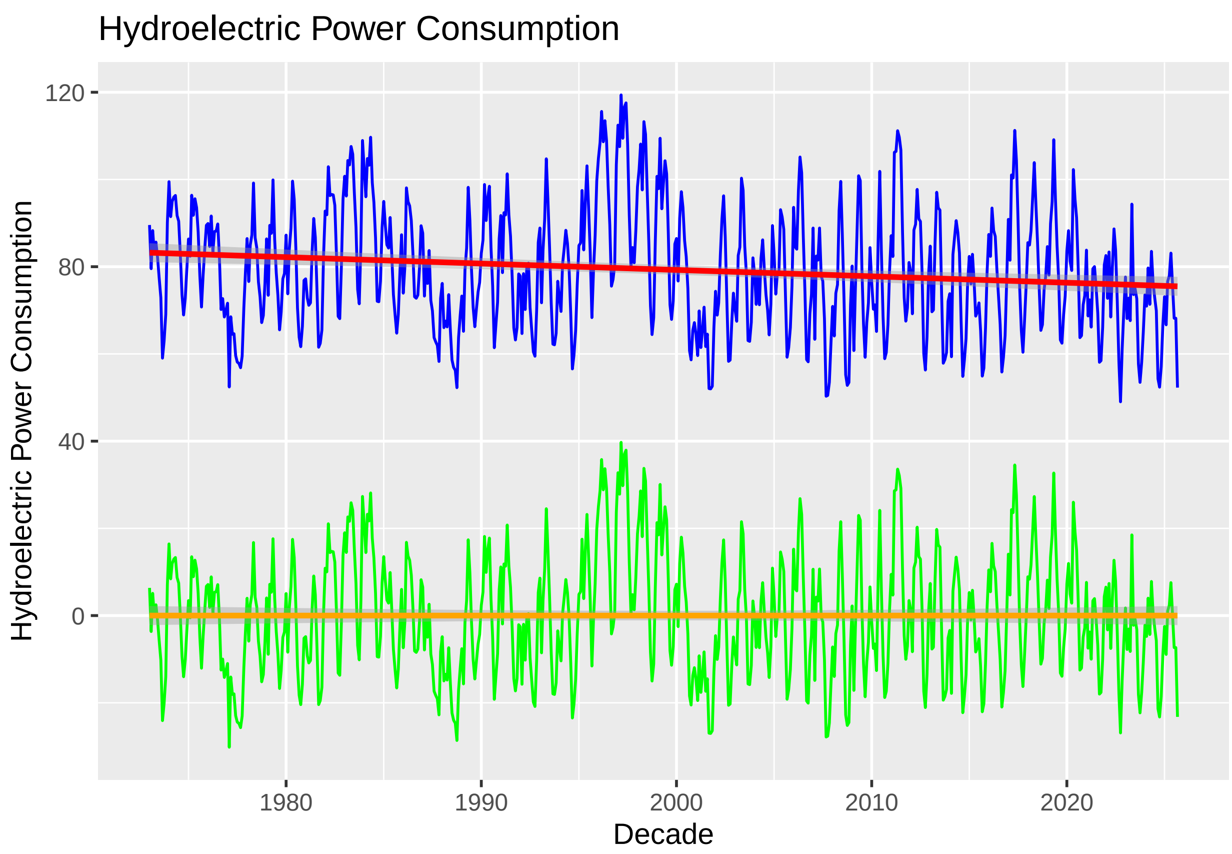
```r
#Renewable: Transfer into ts object
ts_renew_detrend_data <- ts(renew_detrend_data,frequency=12,start=c(1973,1))

#Renewable: Understanding what we did
ggplot(energy_select, aes(x=`Month`, y=`Total Renewable Energy Production`)) +
            geom_line(color="blue") +
            xlab("Decade") +
            ylab(paste0(colnames(energy_select)[(2)],sep="")) +
            #geom_abline(intercept = beta0, slope = beta1, color="red")
            geom_smooth(color="red",method="lm") +
            geom_line(aes(y=renew_detrend_data[[1]]), col="green")+
            geom_smooth(aes(y=renew_detrend_data[[1]]),color="orange",method="lm") +
            labs(title = "Total Renewable Eneergy Production")
```




```r
#Hydroelectric: remove the trend from series
hydro_detrend_data <- energy_select[,3]-(beta00+beta11*t)

#Hydroelectric: note detrend will be a data frame and not a ts object
class(hydro_detrend_data)
```

```
## [1] "data.frame"
```

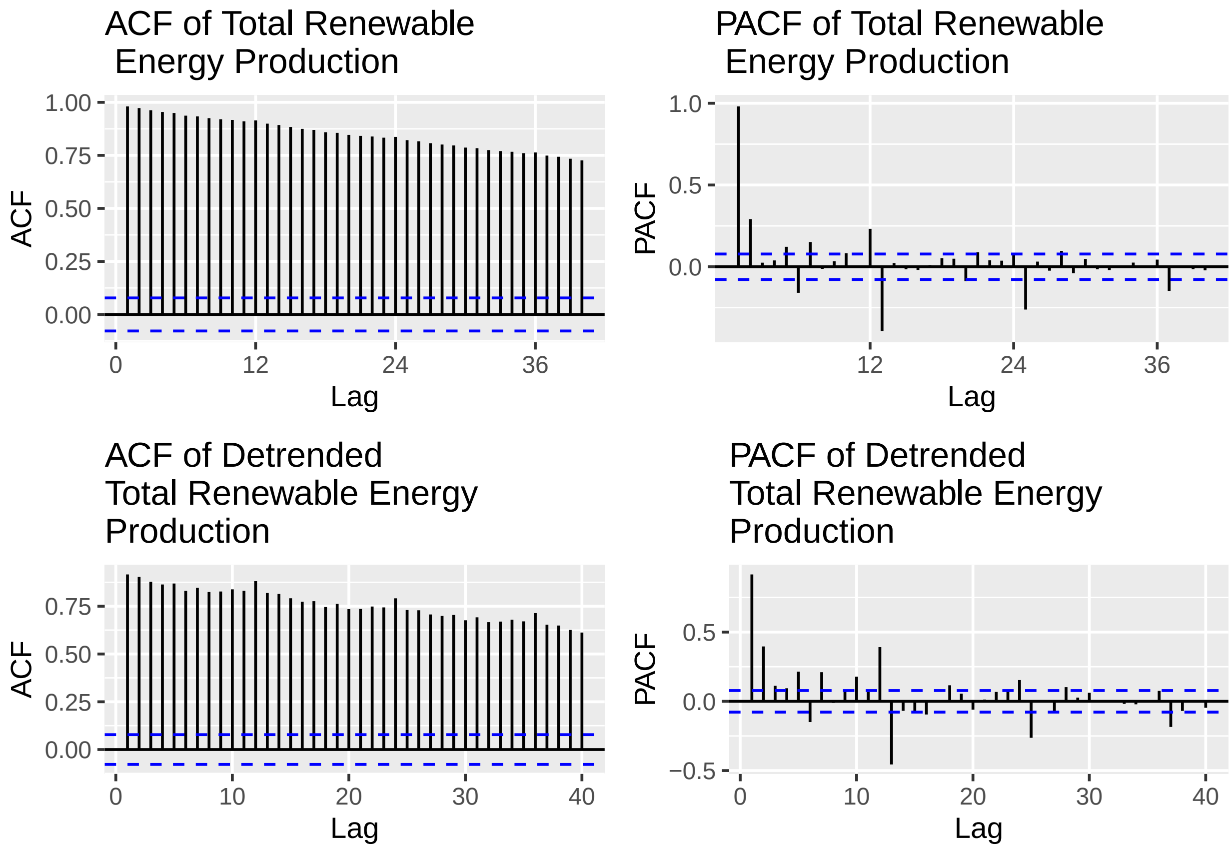
```r
#Hydroelectric: Transfer into ts object
ts_hydro_detrend_data <- ts(hydro_detrend_data,frequency=12,start=c(1973,1))

#Hydroelectric: Understanding what we did
ggplot(energy_select, aes(x=`Month`, y=`Hydroelectric Power Consumption`)) +
            geom_line(color="blue") +
            xlab("Decade") +
            ylab(paste0(colnames(energy_select)[(3)],sep="")) +
            #geom_abline(intercept = beta0, slope = beta1, color="red")
            geom_smooth(color="red",method="lm") +
            geom_line(aes(y=hydro_detrend_data[[1]]), col="green")+
            geom_smooth(aes(y=hydro_detrend_data[[1]]),color="orange",method="lm") +
            labs(title = "Hydroelectric Power Consumption")
```



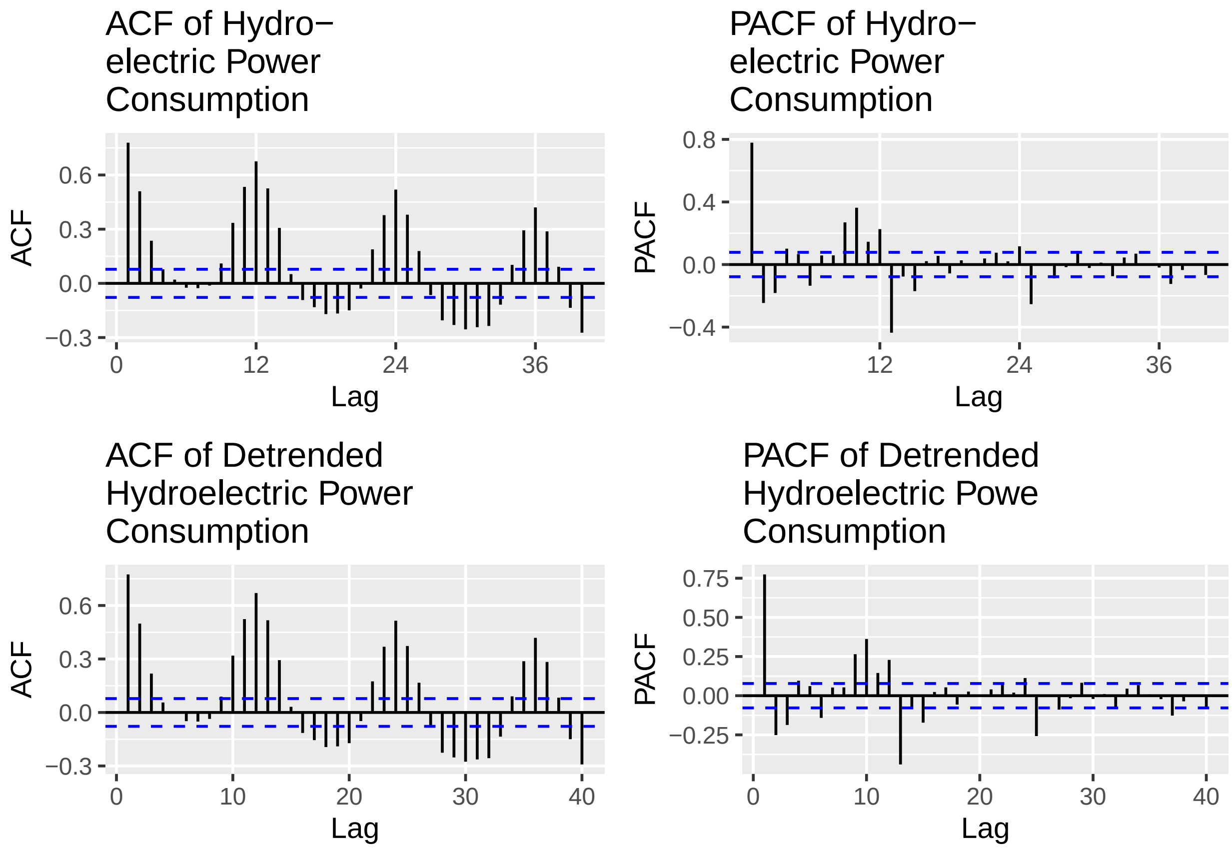
Q5

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?
Compared with Q1, detrending the total renewable energy production weakens its autocorrelations, indicating that much of the earlier persistence was trend-driven. In contrast, the autocorrelation patterns of hydroelectric power consumption remain almost unchanged after detrending, indicating that seasonality has dominant effects. This is likely because hydroelectric power consumption is highly influenced by water availability, which depends on seasonal and weather conditions.

```
detrend_renew_acf_plot = ggAcf(renew_detrend_data, lag.max = 40) +ggtitle("ACF of Detrended\nTotal Renew
detrend_renew_pacf_plot = ggPacf(renew_detrend_data, lag.max = 40) +ggtitle("PACF of Detrended\nTotal Re

detrend_hydro_acf_plot = ggAcf(hydro_detrend_data, lag.max = 40) +ggtitle("ACF of Detrended\nHydroelect
detrend_hydro_pacf_plot = ggPacf(hydro_detrend_data, lag.max = 40) +ggtitle("PACF of Detrended\nHydroel

all_detrend_plot_renew <- plot_grid(renew_acf_plot, renew_pacf_plot, detrend_renew_acf_plot, detrend_re
all_detrend_plot_hydro <- plot_grid(hydro_acf_plot, hydro_pacf_plot, detrend_hydro_acf_plot, detrend_hy

all_detrend_plot_renew
```




```
all_detrend_plot_hydro
```



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 appear to show seasonality, as it shows strong upward trend over time and its ACF remains positive across many lags. However, hydroelectric power consumption shows clear seasonality, as both the time series and the ACF show oscillating patterns over time.

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?

For total renewable energy production, the seasonal means model has low adjusted R-squared of 0.03013, meaning that monthly seasonal effects explain 3% of the variation. The overall p-value of 0.00152 is statistically significant, but only February and September are significant monthly dummy variables. Thus, total renewable energy production does not have a seasonal trend.

For hydroelectric power consumption, the seasonal means model has relatively higher adjusted R-squared of 0.4735, meaning that monthly seasonal effects explain 47% of the variation. The overall p-value (< 2.2e-16) is statistically significant, and most of the monthly dummy variables are significant. Therefore, hydroelectric power consumption has a strong seasonal trend. These results match with my answers in Q6.

```
#Use seasonal means model
#First create the seasonal dummies
renew_dummies <- seasonaldummy(ts_renew_detrend_data)
hydro_dummies <- seasonaldummy(ts_hydro_detrend_data)

#this function only accepts ts object, no need to add one here because date
#object is not a column

#Then fit a linear model to the seasonal dummies
renew_seas_means_model <- lm(ts_renew_detrend_data~renew_dummies)
summary(renew_seas_means_model)
```

```
##
## Call:
## lm(formula = ts_renew_detrend_data ~ renew_dummies)
##
## 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
## renew_dummiesJan    5.840     12.334   0.473   0.6360
## renew_dummiesFeb  -31.525     12.334  -2.556   0.0108 *
## renew_dummiesMar    8.205     12.334   0.665   0.5061
## renew_dummiesApr   -5.400     12.334  -0.438   0.6617
## renew_dummiesMay    8.912     12.334   0.723   0.4703
## renew_dummiesJun   -2.231     12.334  -0.181   0.8565
## renew_dummiesJul    3.114     12.334   0.252   0.8008
## renew_dummiesAug   -5.478     12.334  -0.444   0.6571
## renew_dummiesSep  -31.283     12.334  -2.536   0.0114 *
## renew_dummiesOct  -18.437     12.393  -1.488   0.1373
## renew_dummiesNov  -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
```

```
hydro_seas_means_model <- lm(ts_hydro_detrend_data~hydro_dummies)
summary(hydro_seas_means_model)
```

```
##
## Call:
## lm(formula = ts_hydro_detrend_data ~ hydro_dummies)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -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
## hydro_dummiesJan   4.8900     1.9747   2.476 0.013541 *
## hydro_dummiesFeb  -2.4641     1.9747  -1.248 0.212573
## hydro_dummiesMar   7.0794     1.9747   3.585 0.000364 ***
## hydro_dummiesApr   5.5895     1.9747   2.830 0.004798 **
## hydro_dummiesMay  14.0676     1.9747   7.124 2.92e-12 ***
## hydro_dummiesJun  10.7799     1.9747   5.459 6.93e-08 ***
## hydro_dummiesJul   4.0156     1.9747   2.033 0.042427 *
## hydro_dummiesAug  -5.2952     1.9747  -2.681 0.007525 **
## hydro_dummiesSep -16.5612     1.9747  -8.386 3.37e-16 ***
## hydro_dummiesOct -16.3534     1.9841  -8.242 1.01e-15 ***
## hydro_dummiesNov -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
```

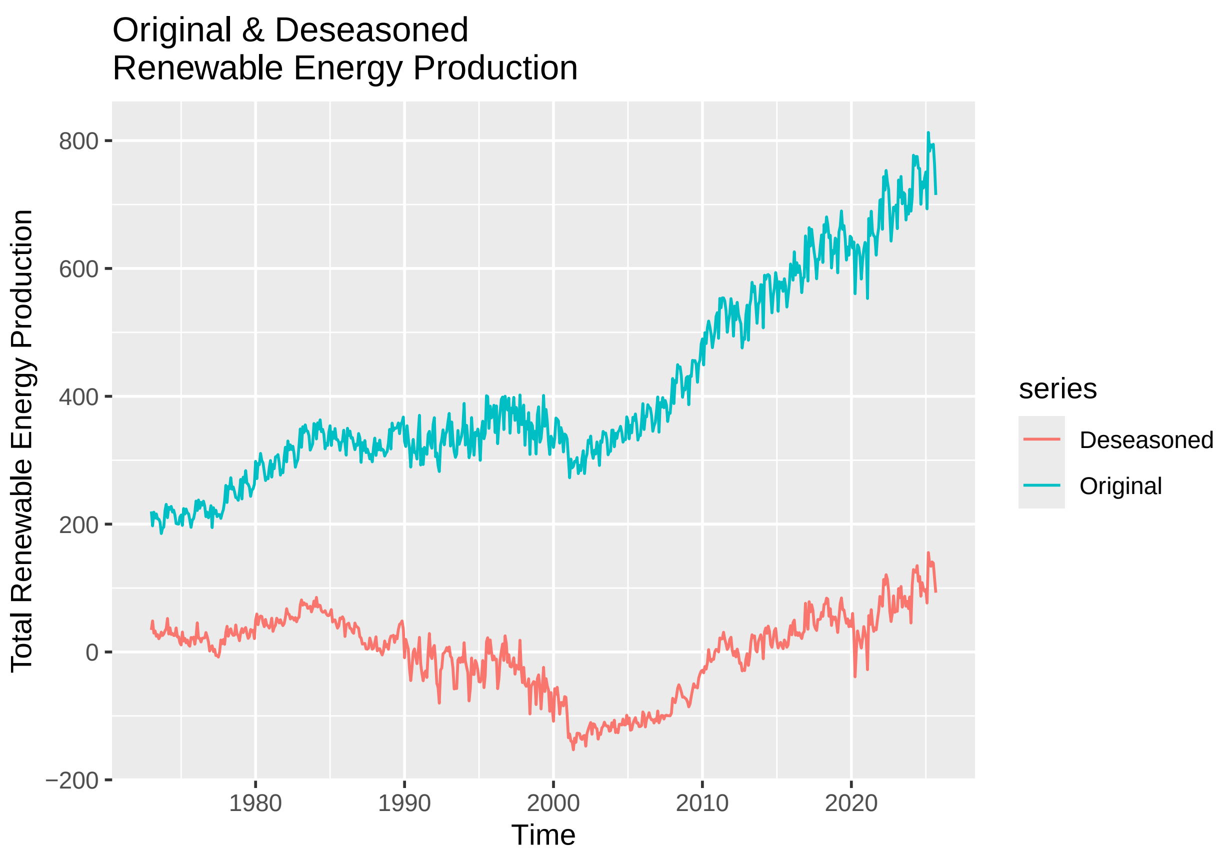
```r
#Look at the regression coefficient. These will be the values of Beta
#Store regression coefficients
renew_int <- renew_seas_means_model$coefficients[1]
renew_coeff <- renew_seas_means_model$coefficients[2:12]

hydro_int <- hydro_seas_means_model$coefficients[1]
hydro_coeff <- hydro_seas_means_model$coefficients[2:12]
```

Q8

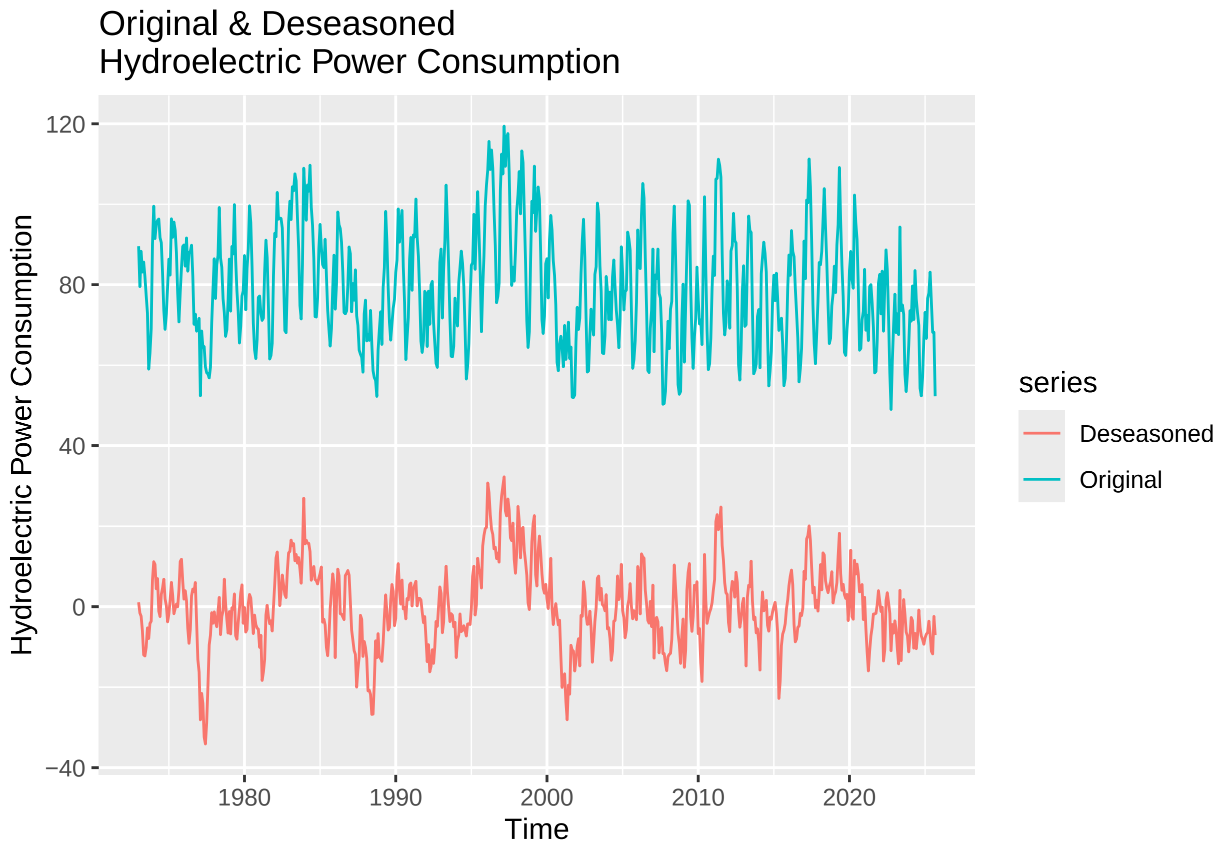
Use the regression coefficients from Q7 to deseason the series. Plot the deseason series and compare with the plots from part Q1. Did anything change?
For total renewable energy production, deseasoning leads to little change in the series, indicating that seasonal effects are weak and do not influence its behavior. However, deseasoning changes the hydroelectric power consumption series. It shows more centered around zero with reduced systematic variation, confirming that its variability is driven by seasonal factors.

```r
#compute seasonal component
renew_seas_comp <- numeric(length(ts_renew_detrend_data))
for(i in 1:length(ts_renew_detrend_data)){
  renew_seas_comp[i] <- (renew_int + renew_coeff %*% renew_dummies[i,])
}

hydro_seas_comp <- numeric(length(ts_hydro_detrend_data))
for(i in 1:length(ts_hydro_detrend_data)){
  hydro_seas_comp[i] <- (hydro_int + hydro_coeff %*% hydro_dummies[i,])
}
```

```r
#Removing seasonal component
renew_deseason_data <- ts_renew_detrend_data - renew_seas_comp
hydro_deseason_data <- ts_hydro_detrend_data - hydro_seas_comp

#Converting to a time series object
ts_deseason_renew_data <- ts(renew_deseason_data,frequency=12,start=c(1973,1))
ts_deseason_hydro_data <- ts(hydro_deseason_data,frequency=12,start=c(1973,1))

#Understanding what we did
autoplot(ts_energy_data[,1], series = 'Original') +
    autolayer(ts_deseason_renew_data, series="Deseasoned")+
    ylab(paste0(colnames(energy_select)[(2)],sep=""))+
    ggtitle('Original & Deseasoned\nRenewable Energy Production')
```




```r
autoplot(ts_energy_data[,2],series = 'Original') +
    autolayer(ts_deseason_hydro_data, series="Deseasoned")+
    ylab(paste0(colnames(energy_select)[(3)],sep=""))+
    ggtitle('Original & Deseasoned\nHydroelectric Power Consumption')
```

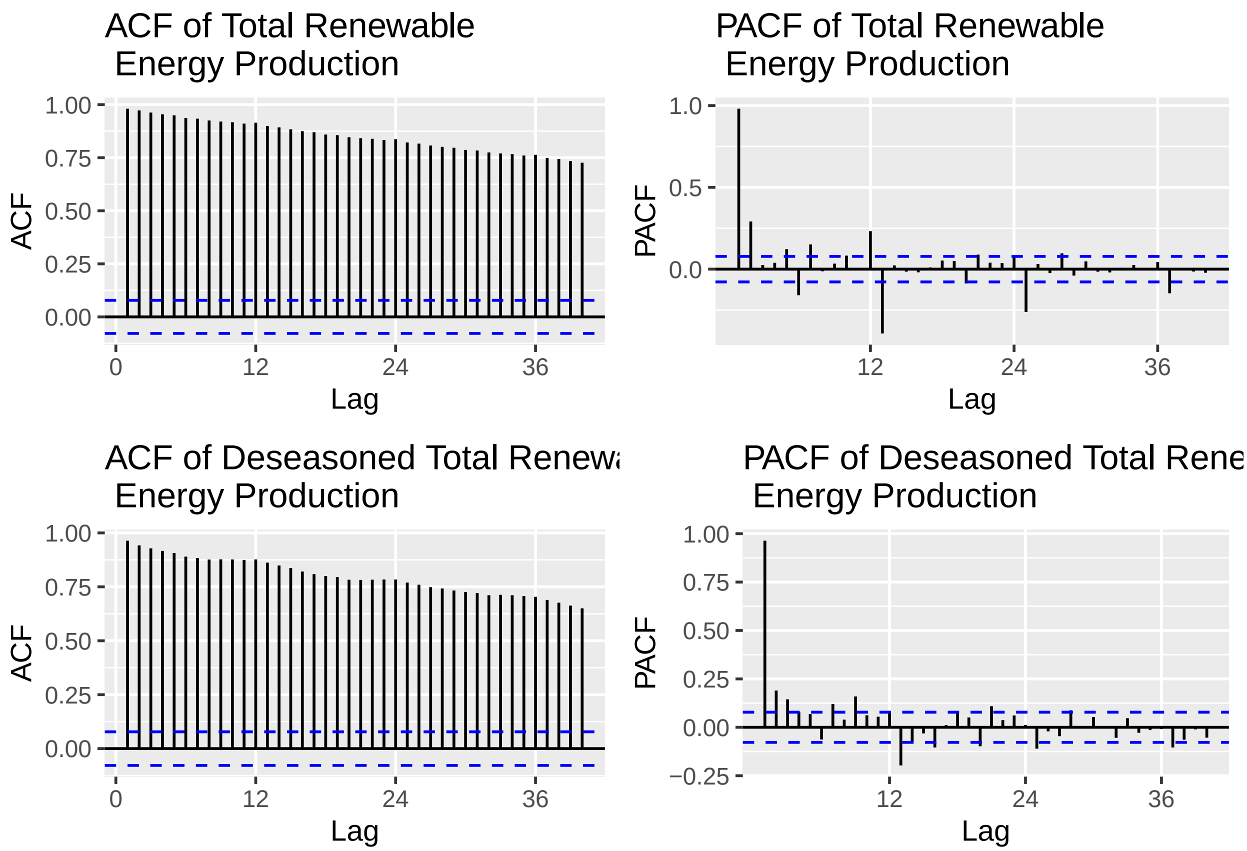


Q9

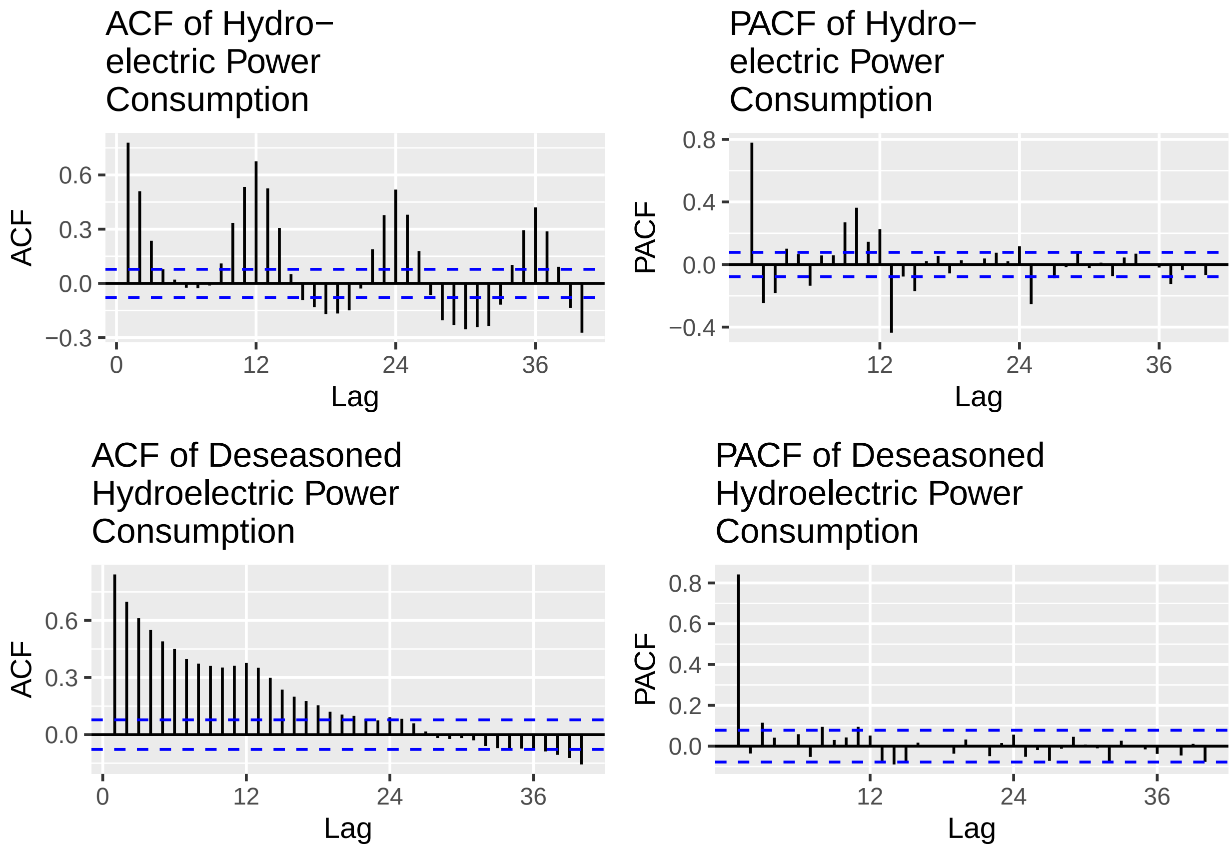
Plot ACF and PACF for the deseason series and compare with the plots from Q1. You may use plot_grid() again to get them side by side. Did the plots change? How?
Deseasoning has little impact on the autocorrelation structure of total renewable energy production, indicating that its dependence is not driven by seasonal effects. However, deseasoning changes the ACF and PACF of hydroelectric power consumption by removing strong seasonal correlations, indicating that seasonality was significant in the original series.

```
#Deseasoned Renewable Energy Plots
des_renew_acf_plot = ggAcf(ts_deseason_renew_data, lag.max = 40) +ggtitle("ACF of Deseasoned Total Renew
des_renew_pacf_plot = ggPacf(ts_deseason_renew_data, lag.max = 40) +ggtitle("PACF of Deseasoned Total Re

#Deseasoned Hydroelectric Power Plots
des_hydro_acf_plot = ggAcf(ts_deseason_hydro_data, lag.max = 40) +ggtitle("ACF of Deseasoned\nHydroelect
des_hydro_pacf_plot = ggPacf(ts_deseason_hydro_data, lag.max = 40) +ggtitle("PACF of Deseasoned\nHydroel

renew_all_plot <- plot_grid(
  renew_acf_plot, renew_pacf_plot,
  des_renew_acf_plot,  des_renew_pacf_plot)

hydro_all_plot <- plot_grid(
  hydro_acf_plot, hydro_pacf_plot,
  des_hydro_acf_plot,  des_hydro_pacf_plot)
```

```
renew_all_plot
```




```
hydro_all_plot
```


ACF of Hydro−
electric Power
Consumption
ACF
0.6
0.3
0.0
−0.3
0
12
24
36
Lag
PACF of Hydro−
electric Power
Consumption
PACF
0.8
0.4
0.0
−0.4
12
24
36
Lag
ACF of Deseasoned
Hydroelectric Power
Consumption
ACF
0.6
0.3
0.0
0
12
24
36
Lag
PACF of Deseasoned
Hydroelectric Power
Consumption
PACF
0.8
0.6
0.4
0.2
0.0
12
24
36
Lag