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

Assignment 3 - Due date 02/03/26

Lauren Shohan

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.\

```r
#Load/install required package here
library(forecast)
```

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

```r
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(lubridate)
```

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

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

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

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

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

```
#extract data
energydata <- read_excel(path=
"/home/guest/TSA_Sp26/Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
skip = 12, sheet="Monthly Data",col_names=FALSE)
```

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

```r
#extract columns
energydata_cols <- read_excel(path=
"/home/guest/TSA_Sp26/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`
```

```r
#assign column names to data
colnames(energydata) <- energydata_cols

#select columns i want - columns 4,5,6
energydata_2col <- energydata[,5:6]

head(energydata_2col)
```

```
## # A tibble: 6 x 2
##   `Total Renewable Energy Production` `Hydroelectric Power Consumption`
##                                 <dbl>                             <dbl>
## 1                                220.                              89.6
## 2                                197.                              79.5
## 3                                219.                              88.3
## 4                                209.                              83.2
## 5                                216.                              85.6
## 6                                208.                              82.1
```

```r
#converting into ts
#starting in 1973 and going monthly, thus freq is 12
ts_energydata <- ts(energydata_2col,start=c(1973,1), frequency=12)
head(ts_energydata)
```

```
##          Total Renewable Energy Production Hydroelectric Power Consumption
## Jan 1973                           219.839                          89.562
## Feb 1973                           197.330                          79.544
## Mar 1973                           218.686                          88.284
## Apr 1973                           209.330                          83.152
## May 1973                           215.982                          85.643
## Jun 1973                           208.249                          82.060
```

##Trend Component

Q1

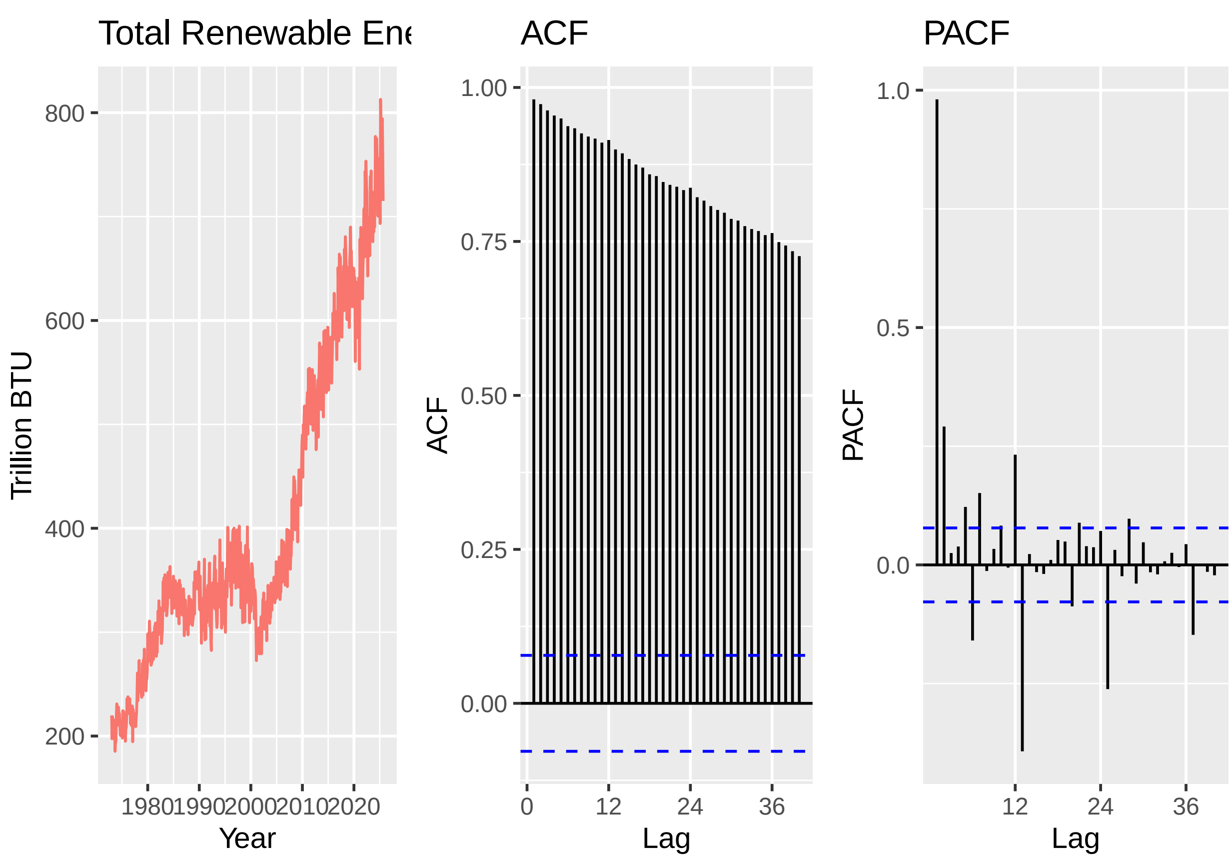
For each series (Total Renewable Production and Hydroelectric Consumption) create three plots arranged in a row (side-by-side): (1) time series plot, (2) ACF, (3) PACF. Use cowplot::plot_grid() to place them in a grid.

```
cowplot::plot_grid(

autoplot(ts_energydata[,1], series = 'Total Renewable Energy Production') +
  xlab("Year") +
  ylab("Trillion BTU") +
  theme(legend.position = "none") +
  ggtitle('Total Renewable Energy Production Time Series'),

autoplot(Acf(ts_energydata[,1], lag.max = 40, plot = FALSE)) + ggtitle('ACF'),


autoplot(Pacf(ts_energydata[,1], lag.max = 40, plot = FALSE)) + ggtitle('PACF'), nrow=1
)
```



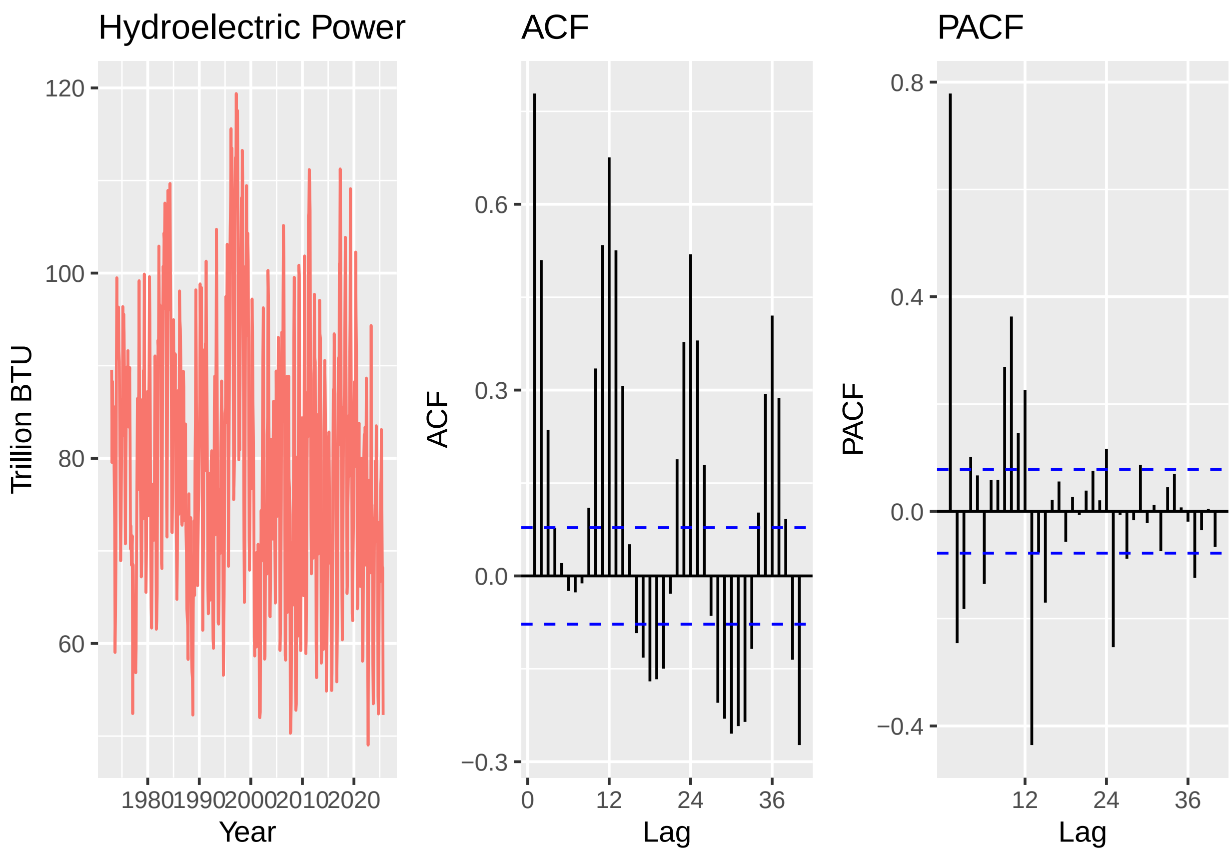

```
plot_grid(

autoplot(ts_energydata[,2], series = 'Hydroelectric Power Consumption') +
  xlab("Year") +
  ylab("Trillion BTU") +
```

```r
  theme(legend.position = "none") +
  ggtitle('Hydroelectric Power Consumption Time Series'),

autoplot(Acf(ts_energydata[,2], lag.max = 40, plot = FALSE)) + ggtitle('ACF'),


autoplot(Pacf(ts_energydata[,2], lag.max = 40, plot = FALSE)) + ggtitle('PACF'), nrow = 1
)
```



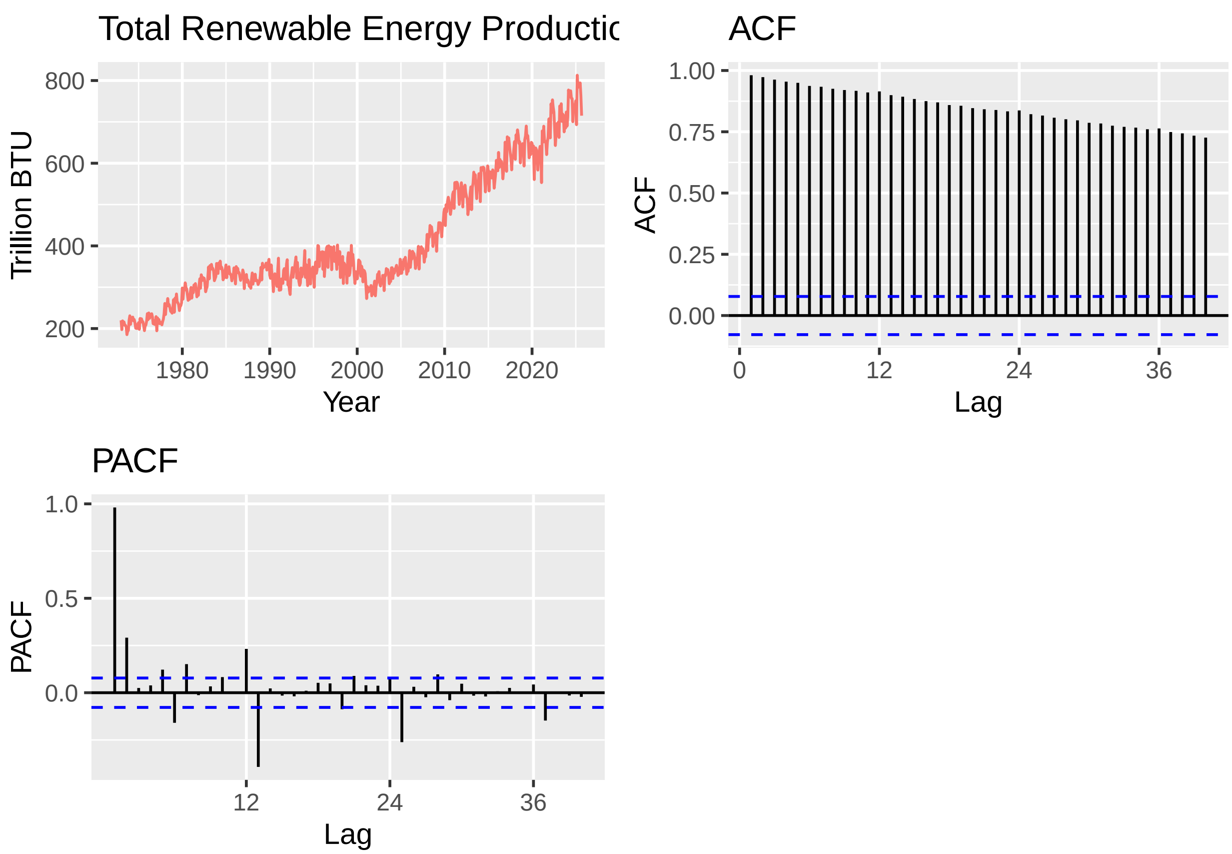
I also made them not three in a row because they squeezed when I did this (above) and it was hard to analyze them like this, I didn't want points off because they weren't directly in a row with one another though.

```r
cowplot::plot_grid(

autoplot(ts_energydata[,1], series = 'Total Renewable Energy Production') +
  xlab("Year") +
  ylab("Trillion BTU") +
  theme(legend.position = "none") +
  ggtitle('Total Renewable Energy Production Time Series'),

autoplot(Acf(ts_energydata[,1], lag.max = 40, plot = FALSE)) + ggtitle('ACF'),


autoplot(Pacf(ts_energydata[,1], lag.max = 40, plot = FALSE)) + ggtitle('PACF')
)
```

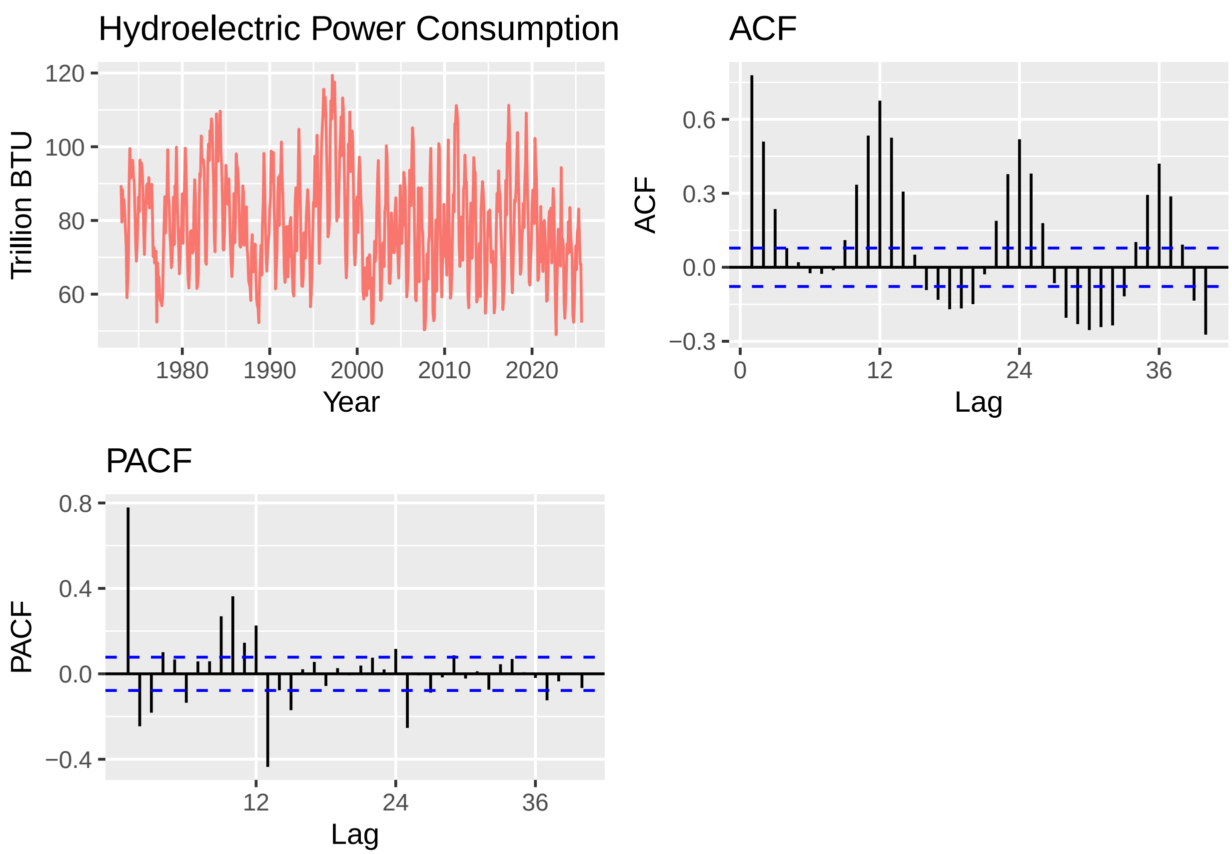



```r
plot_grid(

autoplot(ts_energydata[,2], series = 'Hydroelectric Power Consumption') +
  xlab("Year") +
  ylab("Trillion BTU") +
  theme(legend.position = "none") +
  ggtitle('Hydroelectric Power Consumption Time Series'),

autoplot(Acf(ts_energydata[,2], lag.max = 40, plot = FALSE)) + ggtitle('ACF'),


autoplot(Pacf(ts_energydata[,2], lag.max = 40, plot = FALSE)) + ggtitle('PACF')
)
```



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?

Total Renewable Energy Production appears to have an upward linear trend, with a small dip around year 2000. However, it steadily is rising over the years showcasing an increase of renewable enegy production coming onto the grid and the push for renewables. Hydroelectric power consumption shows a normal seasonal trend one can expect with hydro power, with a strong seasonal component showcases dips during drier periods and higher spikes for excess water periods.

Both ACF graphs have a general downward trend over a certain amount of lags. This slow decay likely indicates strong persistence or memory of the series. While the lags and spikes for the PACF graphs have decreased, the strong spikes at consistent intervals is a good indicator of seasonality.

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(ts_energydata)

#Create vector t
t <- c(1:nobs)
```

```r
#fit linear trend to ts renewable and t
lm_renewable <- lm(ts_energydata[,1]~t)
summary(lm_renewable)
```

```
##
## Call:
## lm(formula = ts_energydata[, 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
#fit linear trend to ts hydro and t
lm_hydro <- lm(ts_energydata[,2]~t)
summary(lm_hydro)
```

```
##
## Call:
## lm(formula = ts_energydata[, 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
#Store reg coefficient
beta0_int_renewable <- as.numeric(lm_renewable$coefficients[1]) #intercept
beta1_slope_renewable <- as.numeric(lm_renewable$coefficients[2]) #slope

beta0_int_hydro <- as.numeric(lm_hydro$coefficients[1]) #intercept
beta1_slope_hydro <- as.numeric(lm_hydro$coefficients[2]) #slope
```

Answer: Renewable Energy Production summary: Intercept is at 171.448 while slope is 0.74999. These summary results show a very strong, positive statistically significant trend over time with a pvalue < 2.2e-16. The R^2 value of 0.8201 shows that around 82% of the variation can be accounted for by the renewable energy production, suggesting a pretty good model fit. Hydro Power Consumption summary: Intercept is at 83.2238 while slope is -0.012199. These summary results show a negative yet statistically significant trend over time with a pvalue 6.547e-05. The R^2 value of 0.02342 shows that only around 23% of the variation can be accounted for by hydro, which is very low suggesting other factors are in play when with the fluctuations of hydro 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.

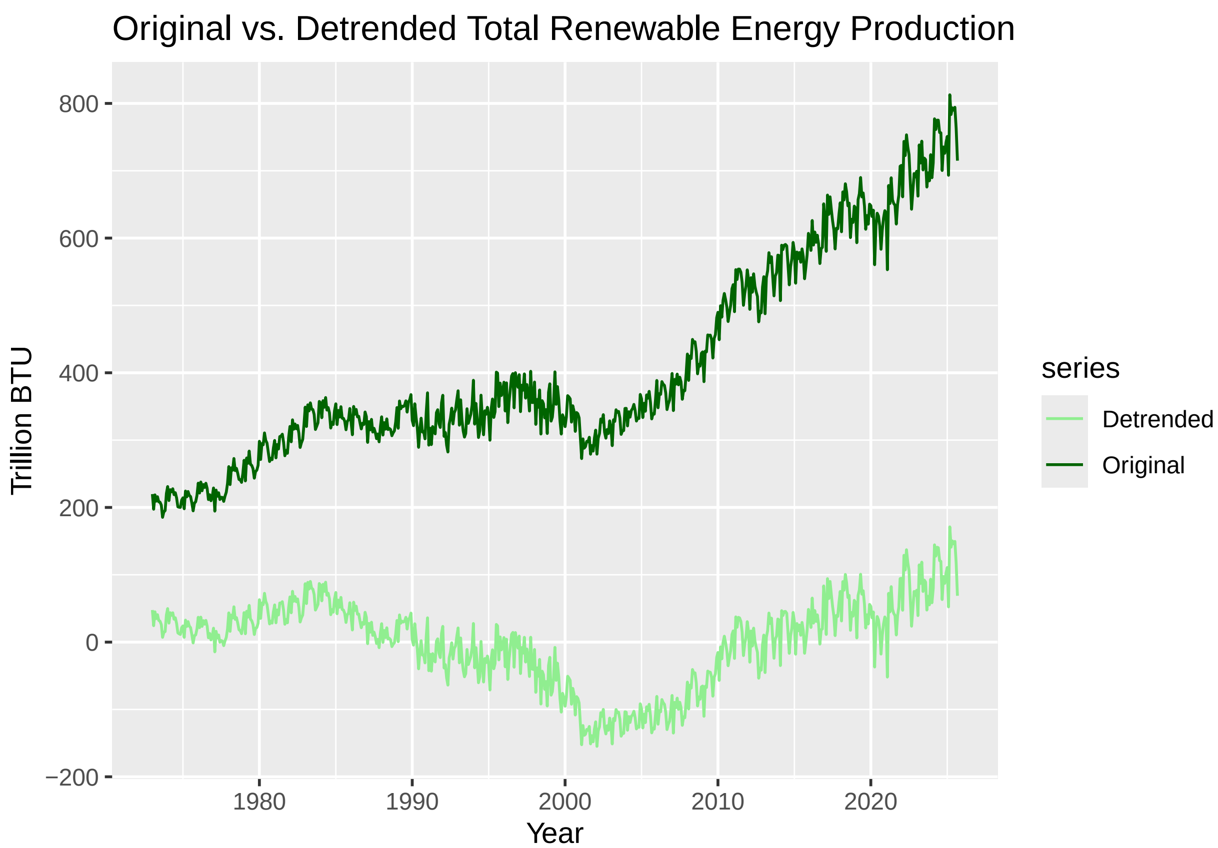
```
# ----- renewable energy ----
#remove the trend from series
detrend_energy_data <- ts_energydata[,1] - (beta0_int_renewable + beta1_slope_renewable*t)
class(detrend_energy_data)
```

```
## [1] "ts"
```

```
#transform to ts
ts_detrend_energy <- ts(detrend_energy_data,frequency = 12,start=c(1973,1))

#plot

autoplot(ts_energydata[,1], series = 'Original') +
  autolayer(ts_detrend_energy, series = 'Detrended') +
  xlab("Year") +
  ylab("Trillion BTU") +
  ggtitle('Original vs. Detrended Total Renewable Energy Production') +
  scale_color_manual(values = c('Original' = 'darkgreen', 'Detrended' = 'lightgreen'))
```



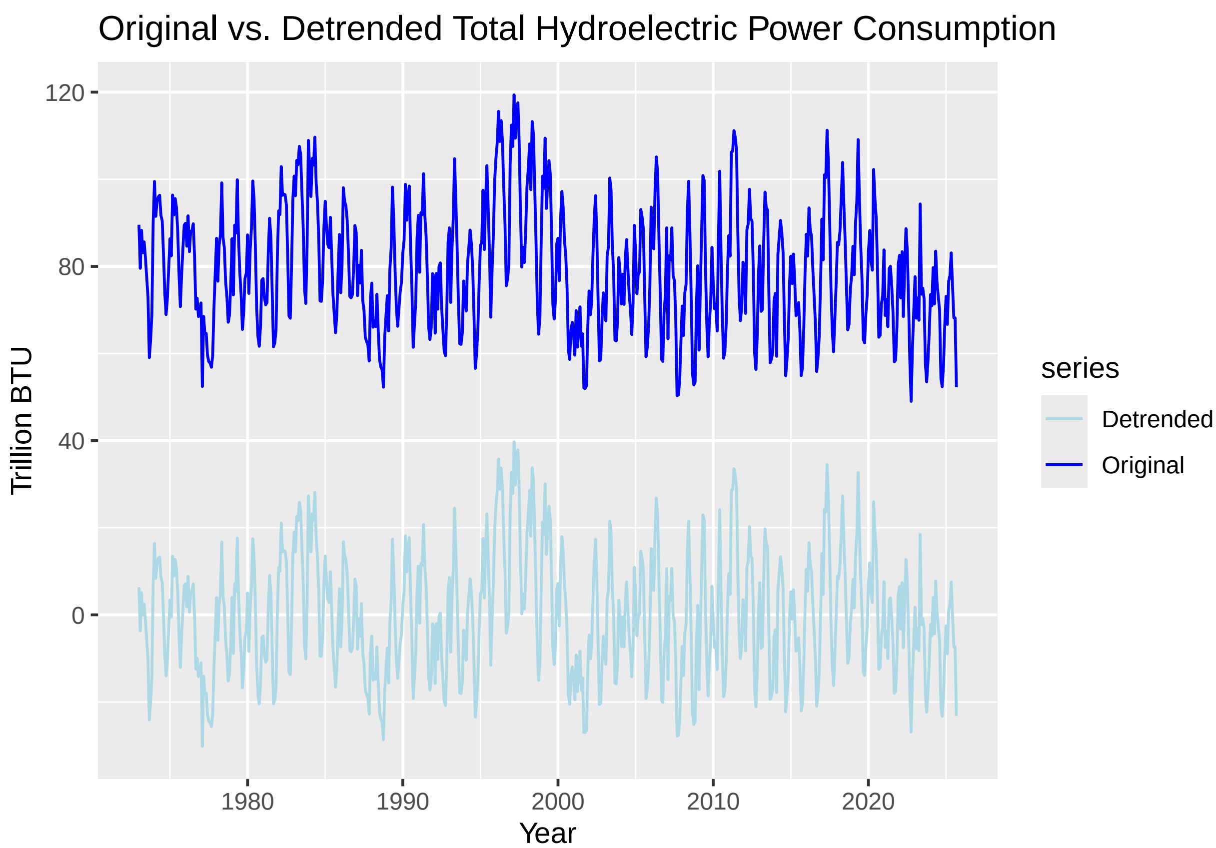

```
# ----- hydro energy ----
#remove the trend from series
detrend_energy_data_hydro <- ts_energydata[,2] - (beta0_int_hydro + beta1_slope_hydro*t)
class(detrend_energy_data_hydro)
```

```
## [1] "ts"
```

```
#transform to ts
ts_detrend_energy_hydro <- ts(detrend_energy_data_hydro,frequency = 12,start=c(1973,1))

#plot

autoplot(ts_energydata[,2], series = 'Original') +
  autolayer(ts_detrend_energy_hydro, series = 'Detrended') +
  xlab("Year") +
  ylab("Trillion BTU") +
  ggtitle('Original vs. Detrended Total Hydroelectric Power Consumption') +
  scale_color_manual(values = c('Original' = 'blue', 'Detrended' = 'lightblue'))
```

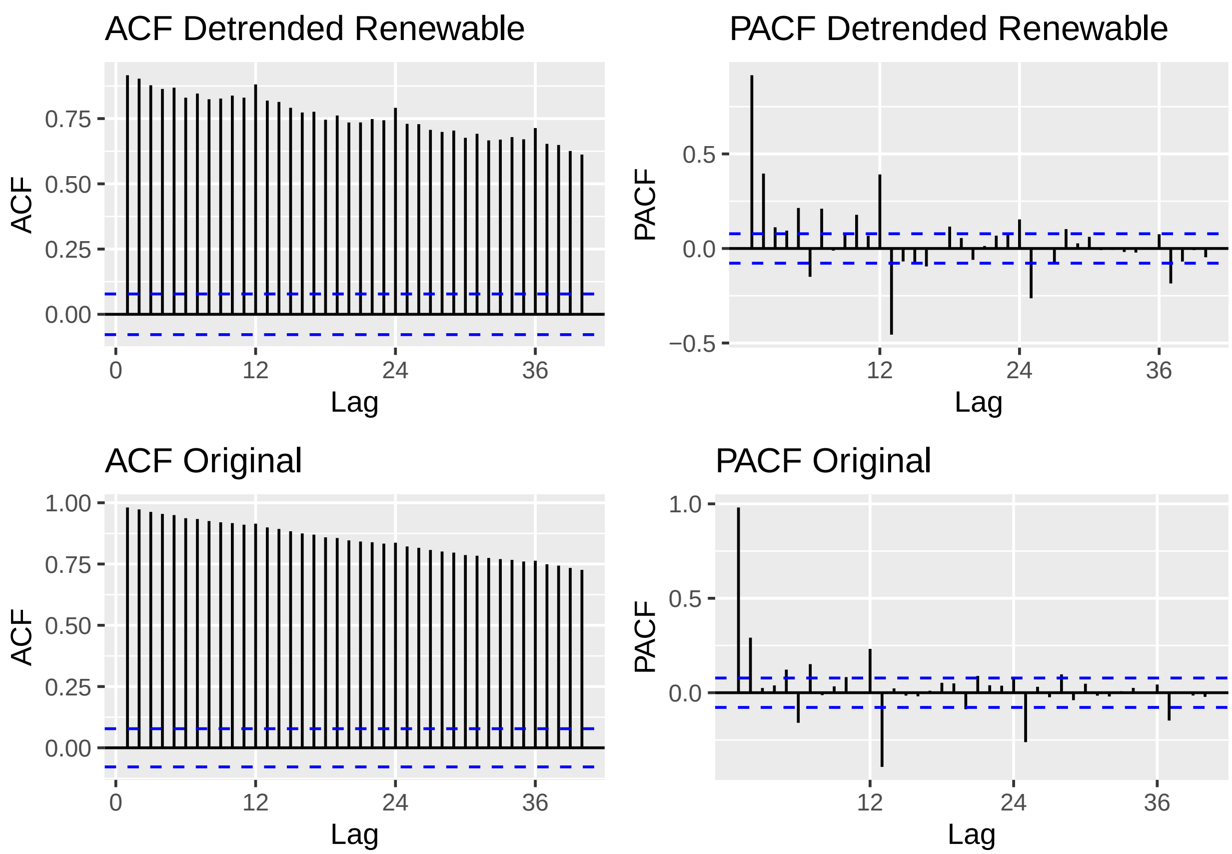


Answer: Both renewable and hydro graphs show a difference after being detrended. Both detrended series are much lower than the original series, but follow similar patterns as the original. However, there is a difference. While hydro detrended seems to follow its original series dip for dip, the renewable energy detrended series is slightly different than the original. Around the year 1985-2000, while the original series increases, the detrended series decreases. Post 2000, both series follow an increasing trend. Both detrended series hover around the 0 trillion btu line, dipping into positive and negative values.

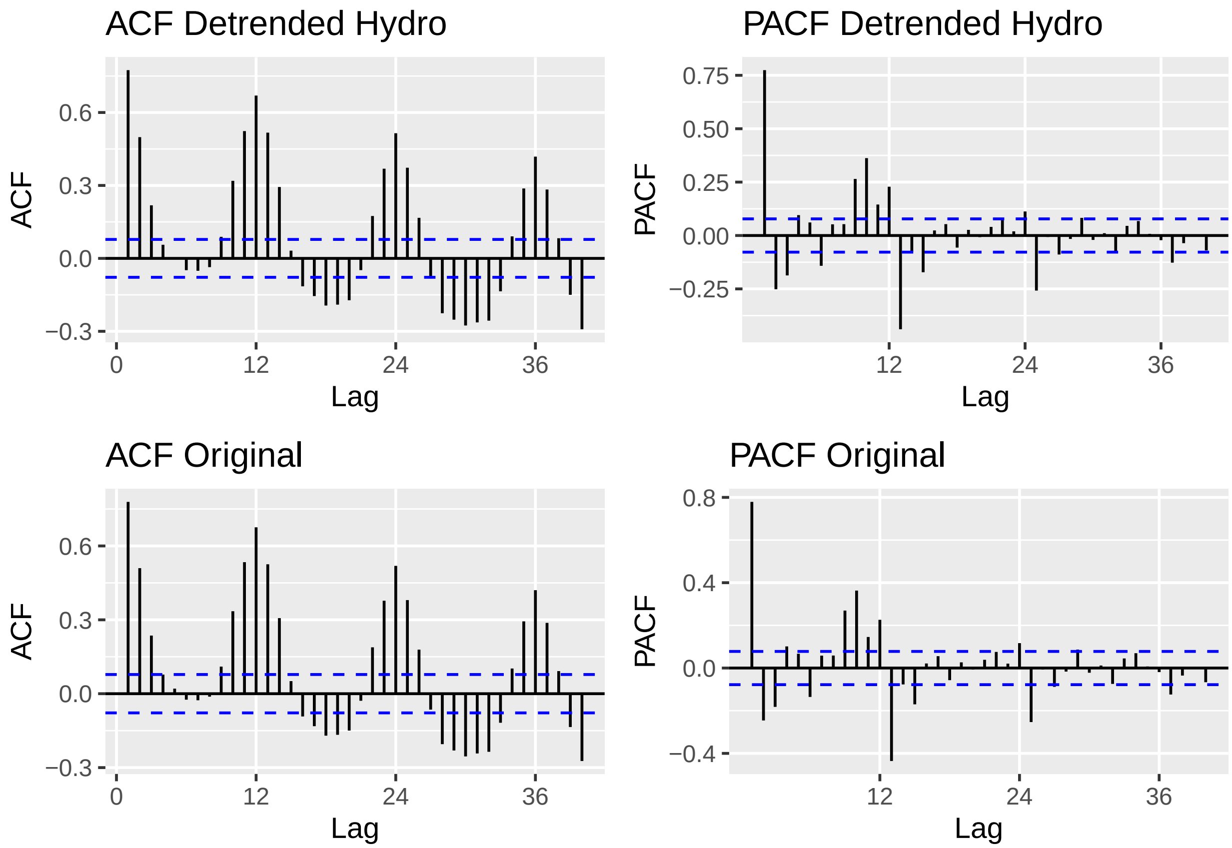
Q5

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?

```
#renewable
plot_grid(
  autoplot(Acf(ts_detrend_energy, lag.max = 40, plot = FALSE)) + ggtitle('ACF Detrended Renewable'), #d
  autoplot(Pacf(ts_detrend_energy, lag.max = 40, plot = FALSE)) + ggtitle('PACF Detrended Renewable'),
  autoplot(Acf(ts_energydata[,1], lag.max = 40, plot = FALSE)) + ggtitle('ACF Original'), #original fro
  autoplot(Pacf(ts_energydata[,1], lag.max = 40, plot = FALSE)) + ggtitle('PACF Original')
  )
```




```
#hydro
plot_grid(
  autoplot(Acf(ts_detrend_energy_hydro, lag.max = 40, plot = FALSE)) + ggtitle('ACF Detrended Hydro'), 
  autoplot(Pacf(ts_detrend_energy_hydro, lag.max = 40, plot = FALSE)) + ggtitle('PACF Detrended Hydro')
  autoplot(Acf(ts_energydata[,2], lag.max = 40, plot = FALSE)) + ggtitle('ACF Original'), #original fro
  autoplot(Pacf(ts_energydata[,2], lag.max = 40, plot = FALSE)) + ggtitle('PACF Original')
  )
```



Answer: After detrending renewable, the ACF decayed much quicker than the original with it almost reaching 0.5 while the original ACF only made it to 0.75. The PACF graphs also became clearer after being detrended, with non signficant spikes increasing and becoming higher in the detrended PACF. However, with the hydro graphs nothing changed between the ACF and PACF detrended and original.

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: Yes they both seem to have seasonality components. The hydro time series shows an obvious visual seasonality component from the time series graph, with dips and spikes correlately to certain months over and over again. The renewable time series potentially showcases seasonality in the PACF graphs with repeating spikes on certain intervals which indicates seasonality.

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?

```r
#----- renewable ---------
#First create the seasonal dummies
dummies_renewable <- seasonaldummy(ts_energydata[,1])

#Then fit a linear model to the seasonal dummies
seas_means_model_renewable <- lm(ts_energydata[,1]~dummies_renewable)
summary(seas_means_model_renewable)
```

```
##
## Call:
## lm(formula = ts_energydata[, 1] ~ dummies_renewable)
##
## 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 ***
## dummies_renewableJan    2.090     29.693   0.070    0.944
## dummies_renewableFeb  -34.524     29.693  -1.163    0.245
## dummies_renewableMar    5.956     29.693   0.201    0.841
## dummies_renewableApr   -6.900     29.693  -0.232    0.816
## dummies_renewableMay    8.162     29.693   0.275    0.784
## dummies_renewableJun   -2.231     29.693  -0.075    0.940
## dummies_renewableJul    3.864     29.693   0.130    0.897
## dummies_renewableAug   -3.978     29.693  -0.134    0.893
## dummies_renewableSep  -29.033     29.693  -0.978    0.329
## dummies_renewableOct  -19.937     29.834  -0.668    0.504
## dummies_renewableNov  -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
```

```r
#----- hydroooooo ---------
#First create the seasonal dummies
dummies_hydro <- seasonaldummy(ts_energydata[,2])

#Then fit a linear model to the seasonal dummies
seas_means_model_hydro <- lm(ts_energydata[,2]~dummies_hydro)
summary(seas_means_model_hydro)
```

```
##
## Call:
## lm(formula = ts_energydata[, 2] ~ dummies_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 ***
## dummies_hydroJan     4.951      2.021   2.449 0.014591 *  
## dummies_hydroFeb    -2.415      2.021  -1.195 0.232608    
## dummies_hydroMar     7.116      2.021   3.520 0.000463 ***
## dummies_hydroApr     5.614      2.021   2.777 0.005649 ** 
## dummies_hydroMay    14.080      2.021   6.965 8.38e-12 ***
## dummies_hydroJun    10.780      2.021   5.333 1.36e-07 ***
## dummies_hydroJul     4.003      2.021   1.980 0.048091 *  
## dummies_hydroAug    -5.320      2.021  -2.632 0.008710 ** 
## dummies_hydroSep   -16.598      2.021  -8.211 1.28e-15 ***
## dummies_hydroOct   -16.329      2.031  -8.040 4.56e-15 ***
## dummies_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
```

> Answer: None of the pvalues for renewable were statistically significant, going month by month none were significant nor was the main summary pvalue of 0.9223. Thus total renewable energy production doesnt show a statistically significant seasonal pattern, which was partially the conclusion i came to previously (i said there was an unclear seasonal pattern). Going month by month with hydro shows that it has strong and statistically significant seasonality component with many months being signficant. This was the conclusion i came to previously. This is typical for hydro, with drier months lower production and more rainfall having higher production during summer/spring months.

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?

```r
#Look at the regression coefficient. These will be the values of Beta
#Store regression coefficients

# ---- renewable ------
beta_int_renewable <- seas_means_model_renewable$coefficients[1]
beta_coeff_renewable <- seas_means_model_renewable$coefficients[2:12]

# ------- hydro ---------
beta_int_hydro <- seas_means_model_hydro$coefficients[1]
beta_coeff_hydro <- seas_means_model_hydro$coefficients[2:12]
```

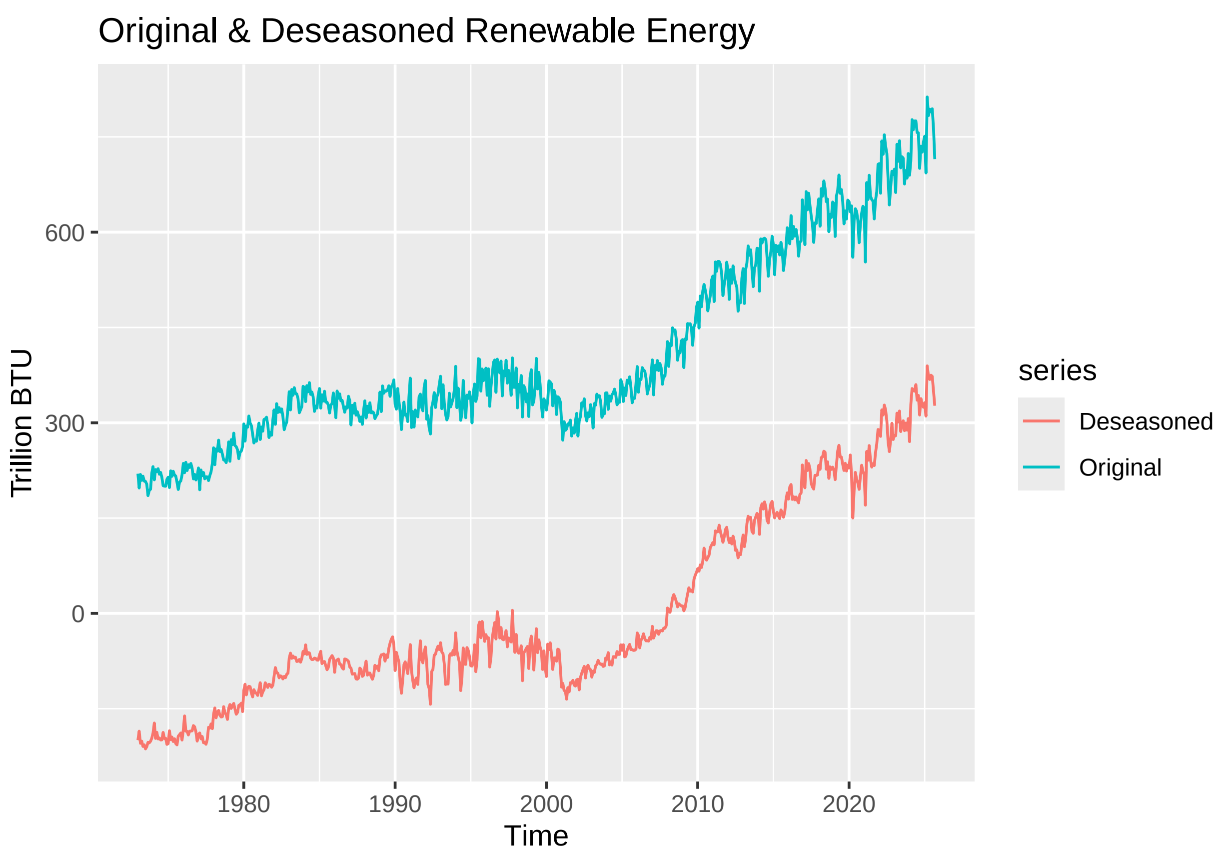
```r
#compute seasonal component
# ------- renewable -------
renewable_seas_comp <- array(0,nobs)
for(i in 1:nobs){
  renewable_seas_comp[i] <- (beta_int_renewable+beta_coeff_renewable %*% dummies_renewable[i,])
```

```r
}

#Removing seasonal component
deseason_renewable_data <- ts_energydata[,1]-renewable_seas_comp

#Converting to a time series object
ts_deseason_renewable_data <- ts(deseason_renewable_data,frequency=12,start=c(1973,1))

#Understanding what we did
autoplot(ts_energydata[,1], series = 'Original') +
  autolayer(ts_deseason_renewable_data, series="Deseasoned") +
  ylab("Trillion BTU") +
  ggtitle('Original & Deseasoned Renewable Energy')
```



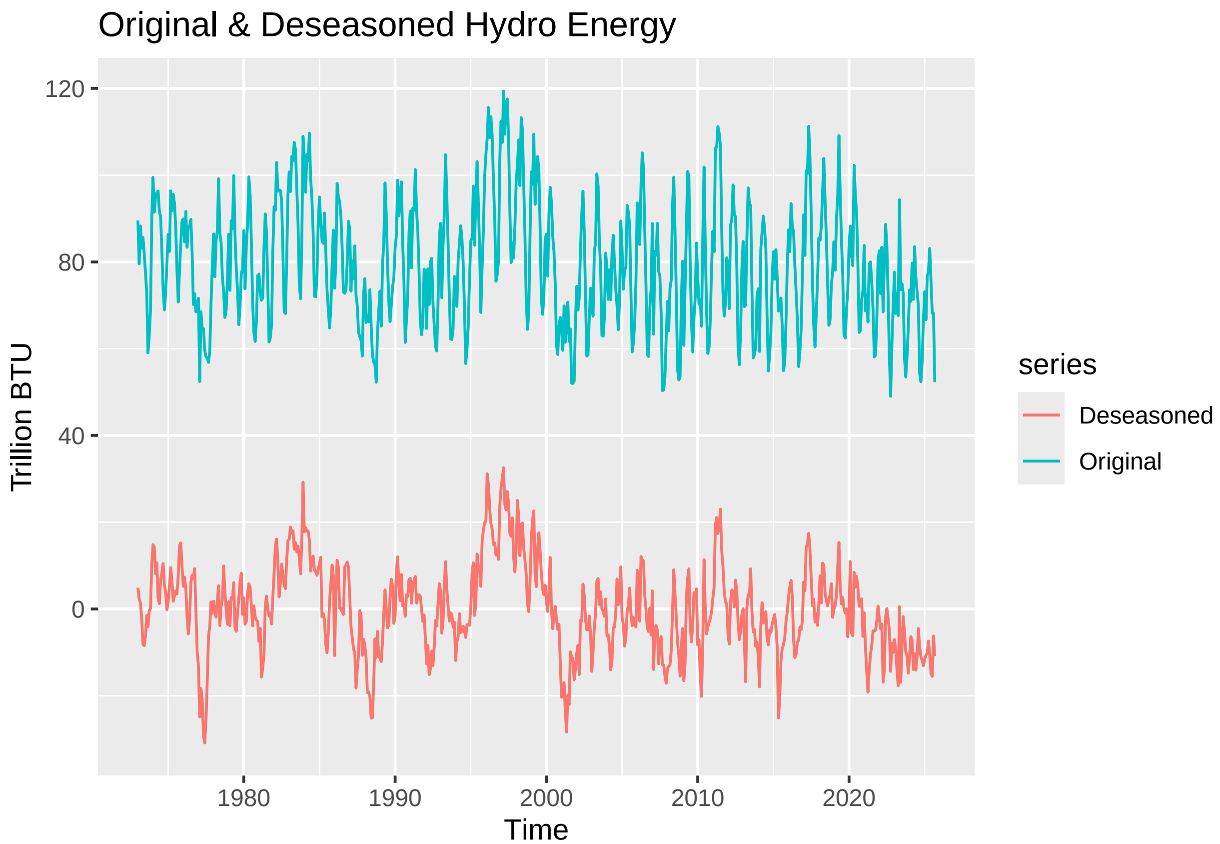

```r
#compute seasonal component
# ------- hydro -------
hydro_seas_comp <- array(0,nobs)
for(i in 1:nobs){
  hydro_seas_comp[i] <- (beta_int_hydro+beta_coeff_hydro %*% dummies_hydro[i,])
}

#Removing seasonal component
deseason_hydro_data <- ts_energydata[,2]-hydro_seas_comp

#Converting to a time series object
```

```r
ts_deseason_hydro_data <- ts(deseason_hydro_data,frequency=12,start=c(1973,1))

#Understanding what we did
autoplot(ts_energydata[,2], series = 'Original') +
    autolayer(ts_deseason_hydro_data, series="Deseasoned")+
     ylab("Trillion BTU") + ggtitle('Original & Deseasoned Hydro Energy')
```

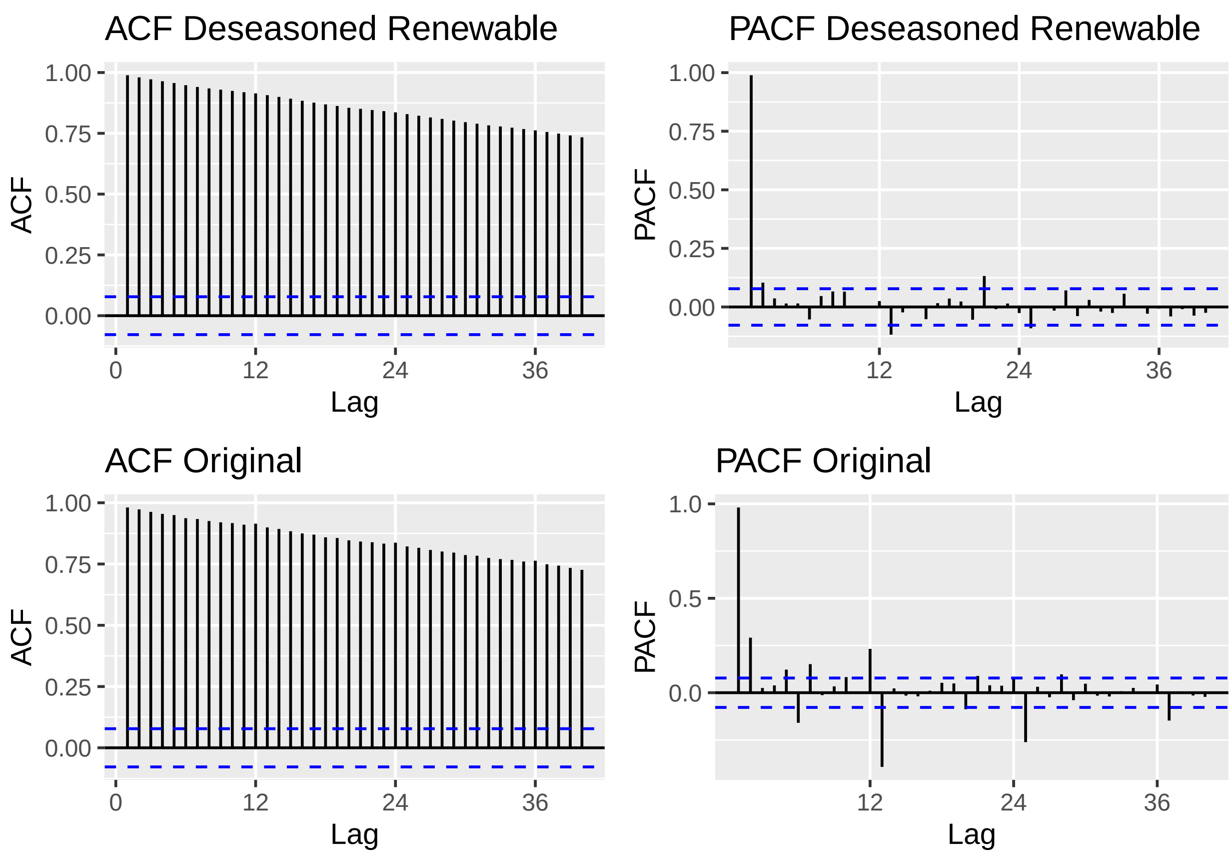


> Answer: Deseasoning had no effect on renewable production because the seasonal means model was not statistically significant thus it is the same graph just much lower. However, the deseasoned hydro graph does change and becomes smoother and easier to follow visually with reduced oscillations. Deseasoning hydro reduces these fluctuations which affirms my conclusion of a strong seasonality component.

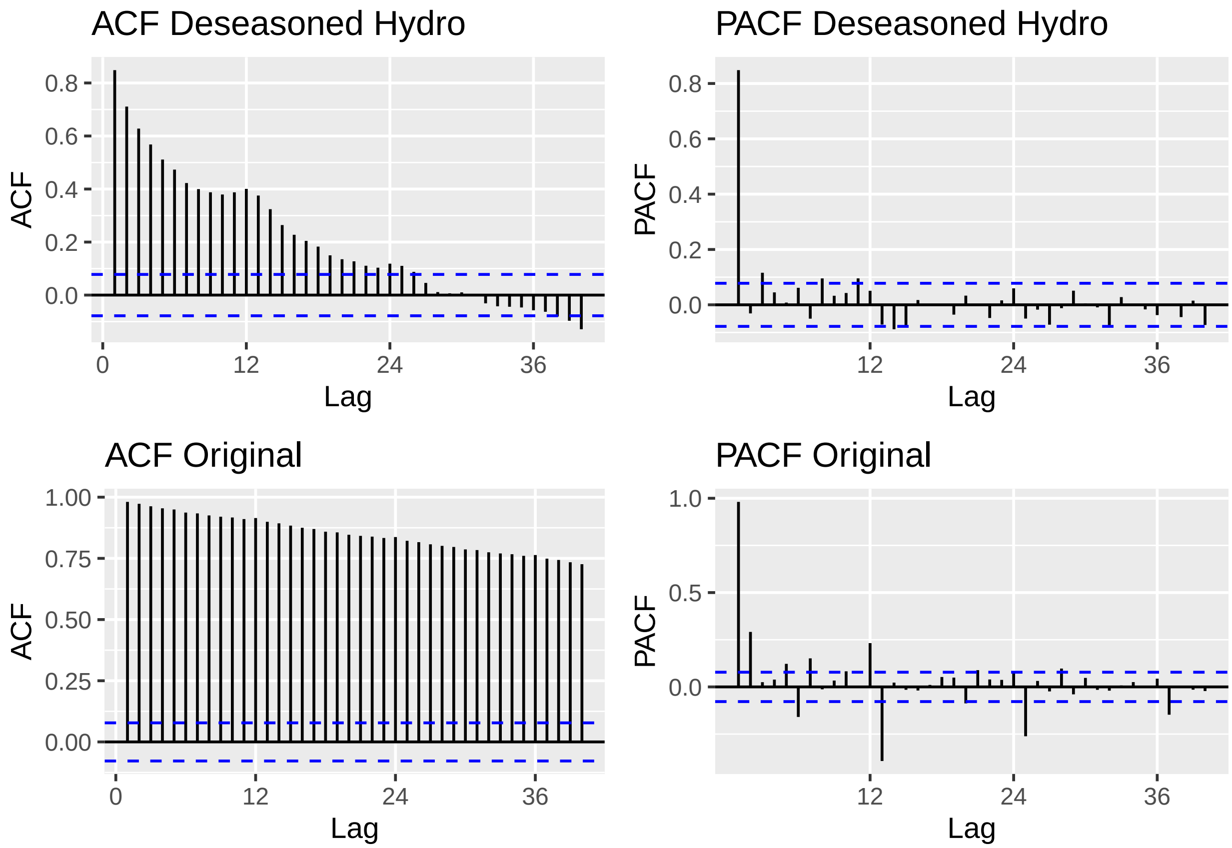
Q9

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?

```r
#renewable
plot_grid(
  autoplot(Acf(ts_deseason_renewable_data, lag.max = 40, plot = FALSE)) + ggtitle('ACF Deseasoned Renewa
  autoplot(Pacf(ts_deseason_renewable_data, lag.max = 40, plot = FALSE)) + ggtitle('PACF Deseasoned Rene
  autoplot(Acf(ts_energydata[,1], lag.max = 40, plot = FALSE)) + ggtitle('ACF Original'),
  autoplot(Pacf(ts_energydata[,1], lag.max = 40, plot = FALSE)) + ggtitle('PACF Original')
  )
```




```r
#renewable
plot_grid(
  autoplot(Acf(ts_deseason_hydro_data, lag.max = 40, plot = FALSE)) + ggtitle('ACF Deseasoned Hydro'),
  autoplot(Pacf(ts_deseason_hydro_data, lag.max = 40, plot = FALSE)) + ggtitle('PACF Deseasoned Hydro')
  autoplot(Acf(ts_energydata[,1], lag.max = 40, plot = FALSE)) + ggtitle('ACF Original'),
  autoplot(Pacf(ts_energydata[,1], lag.max = 40, plot = FALSE)) + ggtitle('PACF Original')
  )
```



Answer: For renewable, the ACF graph did not change at all but the PACF deseasoned graph did change. The spikes decreased and basically all of them became insignificant. With hydro, both the ACF and PACF graph changed drastically. The ACF original showed a smooth downward trend, while the deseasoned one became bumpy, with a major wave in the middle and ultimately turned negative. The PACF deseasoned followed a similar trend to the renewable, with the PACF spikes decreased in height very signficantly compared to the original.