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

Assignment 3 - Due date 02/03/26

Nicole Gutkowski

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)
```

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

```
library(tseries)
library(Kendall)
library(cowplot)
```

```r
library(ggplot2)
library(readxl)
library(openxlsx)
library(lubridate)
```

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

## The following object is masked from 'package:cowplot':
##
##     stamp

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

```r
energy_data1 =
  read_xlsx(
    path ="./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
#Now let's extract the column names from row 11
read_col_names <-
  read_excel(
    path="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
    skip = 10,n_max = 1, sheet="Monthly Data",col_names=FALSE)
```

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

```
## * `` -> `...8`
## * `` -> `...9`
## * `` -> `...10`
## * `` -> `...11`
## * `` -> `...12`
## * `` -> `...13`
## * `` -> `...14`
```

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

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

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

##Trend Component

Q1

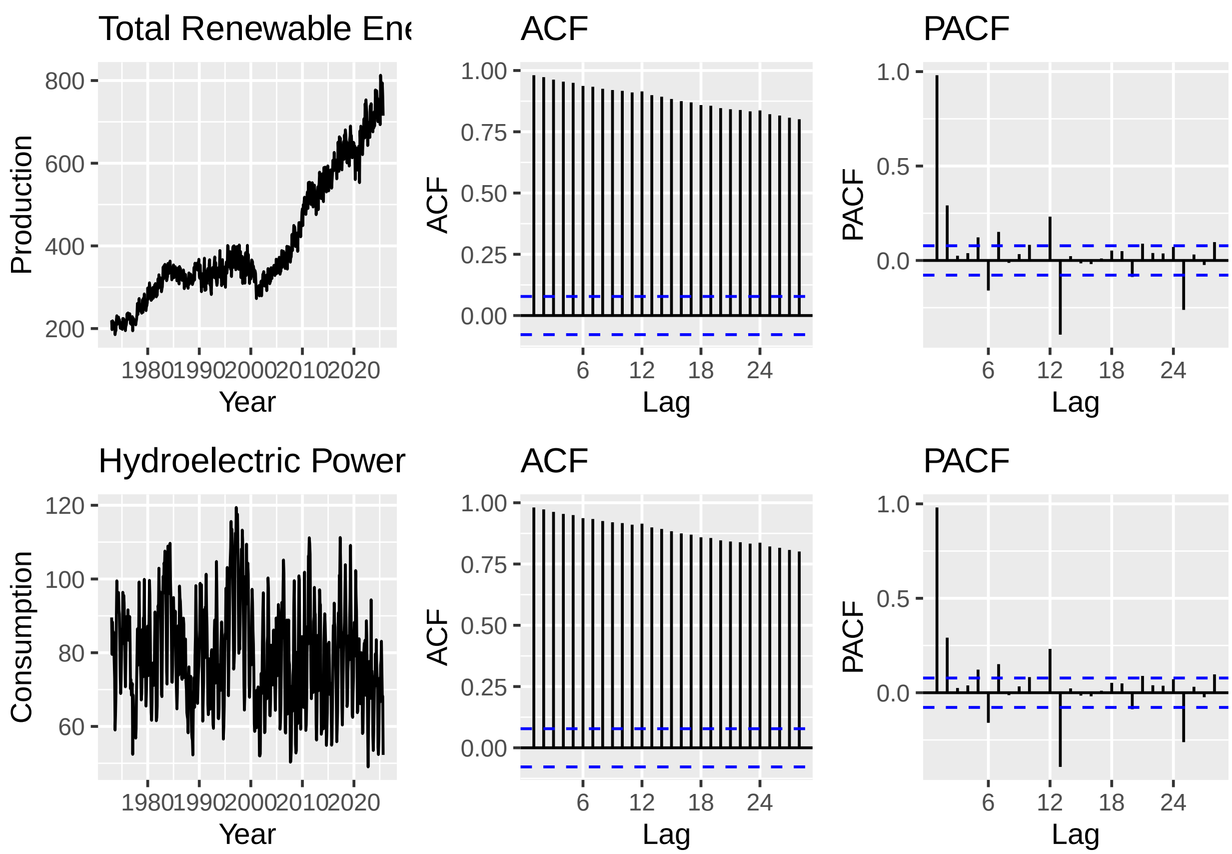
For each series (Total Renewable Production and Hydroelectric Consumption) create three plots arranged in a row (side-by-side): (1) time series plot, (2) ACF, (3) PACF. Use cowplot::plot_grid() to place them in a grid.

```
subset_df = energy_data1[,c("Month",
                            "Total Renewable Energy Production",
                            "Hydroelectric Power Consumption")]
subset_df$Month <- as.Date(subset_df$Month)

head(subset_df)
```

```
## # A tibble: 6 x 3
##   Month      `Total Renewable Energy Production` Hydroelectric Power Consumpti~1
##   <date>                                   <dbl>                           <dbl>
## 1 1973-01-01                                220.                            89.6
## 2 1973-02-01                                197.                            79.5
## 3 1973-03-01                                219.                            88.3
```

```
## 4 1973-04-01                                  209.                            83.2
## 5 1973-05-01                                  216.                            85.6
## 6 1973-06-01                                  208.                            82.1
## # i abbreviated name: 1: `Hydroelectric Power Consumption`
```

```r
ts_data = ts(subset_df[,2:3],frequency=12, start = c(1973,1))


plot_ts_tren <- autoplot(ts_data[,1]) +
  labs(title = "Total Renewable Energy Production",
       x = "Year", y = "Production")

acf_tren  <- Acf(ts_data[,1], plot = FALSE)
pacf_tren <- Pacf(ts_data[,1], plot = FALSE)

plot_acf_tren <- autoplot(acf_tren) + labs(title = "ACF",
       x = "Lag", y = "ACF")
plot_pacf_tren <- autoplot(pacf_tren) + labs(title = "PACF",
       x = "Lag", y = "PACF")


plot_ts_hydro <- autoplot(ts_data[,2]) +
  labs(title = "Hydroelectric Power Consumption",
       x = "Year", y = "Consumption")

acf_hydro  <- Acf(ts_data[,2], plot = FALSE)
pacf_hydro <- Pacf(ts_data[,2], plot = FALSE)

plot_acf_hydro <- autoplot(acf_tren) + labs(title = "ACF",
       x = "Lag", y = "ACF")
plot_pacf_hydro <- autoplot(pacf_tren) + labs(title = "PACF",
       x = "Lag", y = "PACF")

all_plots <- plot_grid(plot_ts_tren, plot_acf_tren, plot_pacf_tren, plot_ts_hydro, plot_acf_hydro, plot_

all_plots
```



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: The series Total Renewable Energy Production appears to have an upwards linear trend with potential seasonality. The series Hydroelectric Power Comsumption appears to have a strong seasonal pattern.

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.

```
#Remove last year by replacing current data frame
nobs <- nrow(subset_df)

shortened_df <- subset_df[1:(nobs-9),]


#Tail again to check if the rows were correctly removed
tail(shortened_df)
```

```
## # A tibble: 6 x 3
##   Month      `Total Renewable Energy Production` Hydroelectric Power Consumpti~1
##   <date>                                   <dbl>                           <dbl>
## 1 2024-07-01                                757.                            73.0
## 2 2024-08-01                                756.                            69.9
## 3 2024-09-01                                700.                            54.3
## 4 2024-10-01                                735.                            52.4
## 5 2024-11-01                                726.                            57.1
## 6 2024-12-01                                742.                            66.6
## # i abbreviated name: 1: `Hydroelectric Power Consumption`
```

because the linear regression is not supposed to be a time series object!! So trying to fix the end of the subset df so that the rows go to month 12/ no overlap?

```r
nobs <- nrow(shortened_df)
t <- c(1:nobs)

energy_data = as.data.frame(shortened_df)
ts_energy_data = ts(energy_data[,2:3],frequency=12, start = c(1973,1))


#linear trend for Total Renew
linear_trend_tren <- lm(shortened_df[[2]]~t)
#regressing over the time vector just created
summary(linear_trend_tren)
```

```
##
## Call:
## lm(formula = shortened_df[[2]] ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -152.38  -38.35   12.95   41.57  151.49
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 175.08397    5.02585   34.84   <2e-16 ***
## t             0.73268    0.01393   52.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 62.7 on 622 degrees of freedom
## Multiple R-squared:  0.8164, Adjusted R-squared:  0.8161
## F-statistic:  2765 on 1 and 622 DF,  p-value: < 2.2e-16
```

```r
#store the reg coefficients
beta1_tren <- as.numeric(linear_trend_tren$coefficients[1]) #intercept
beta2_tren <- as.numeric(linear_trend_tren$coefficients[2]) #slope

#linear trend for Hydro Power
```

```r
linear_trend_hydro <- lm(shortened_df[[3]]~t)
#regressing over the time vector just created
summary(linear_trend_hydro)
```

```
##
## Call:
## lm(formula = shortened_df[[3]] ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.093 -10.277  -0.700   9.073  39.675
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 83.096738   1.122912  74.001  < 2e-16 ***
## t           -0.011596   0.003113  -3.725 0.000213 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.01 on 622 degrees of freedom
## Multiple R-squared:  0.02182,    Adjusted R-squared:  0.02025
## F-statistic: 13.87 on 1 and 622 DF,  p-value: 0.0002132
```

```r
#store the reg coefficients
beta1_hydro <- as.numeric(linear_trend_hydro$coefficients[1]) #intercept
beta2_hydro <- as.numeric(linear_trend_hydro$coefficients[2]) #slope
```

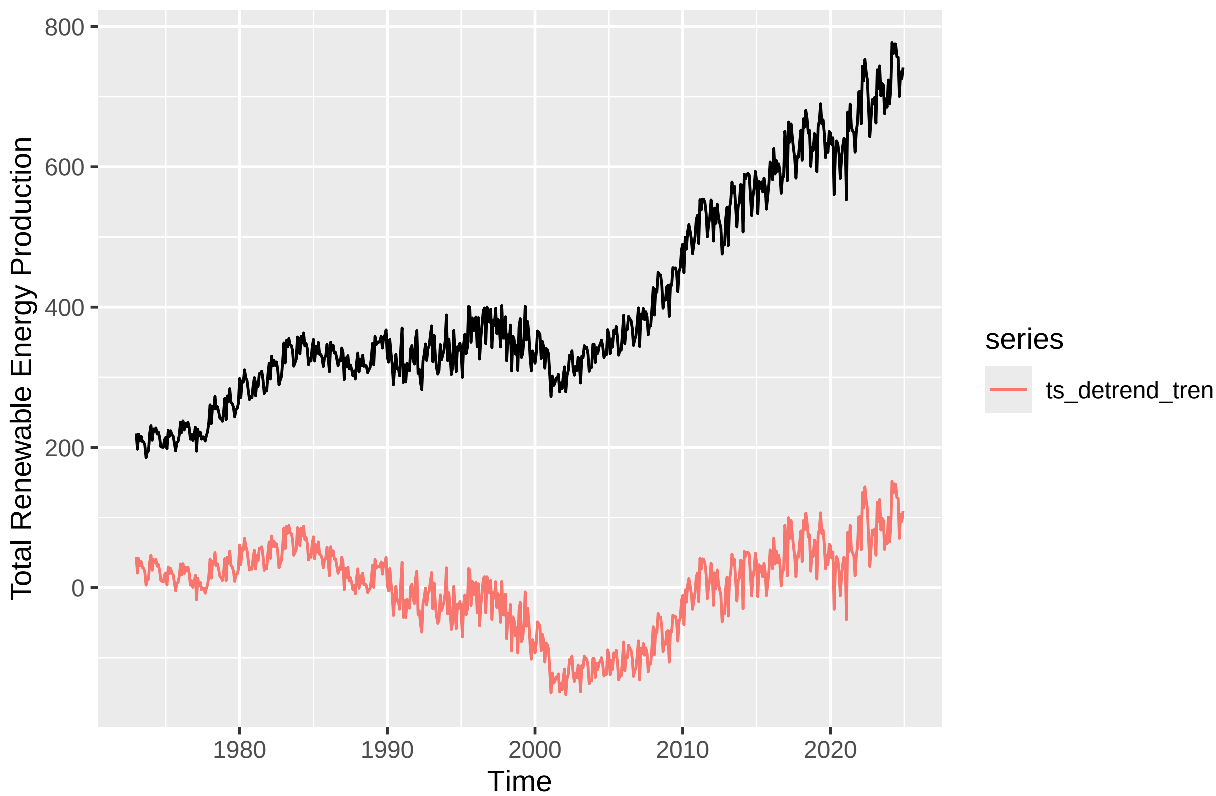
Q4

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.

```r
#Total Renewable
#remove the trend from series total series
detrend_tren_data <- shortened_df[[2]] - (beta1_tren + beta2_tren*t)
#subtracting the equation (with the stored coeffs and the t value)
class(detrend_tren_data)
```

```
## [1] "numeric"
```

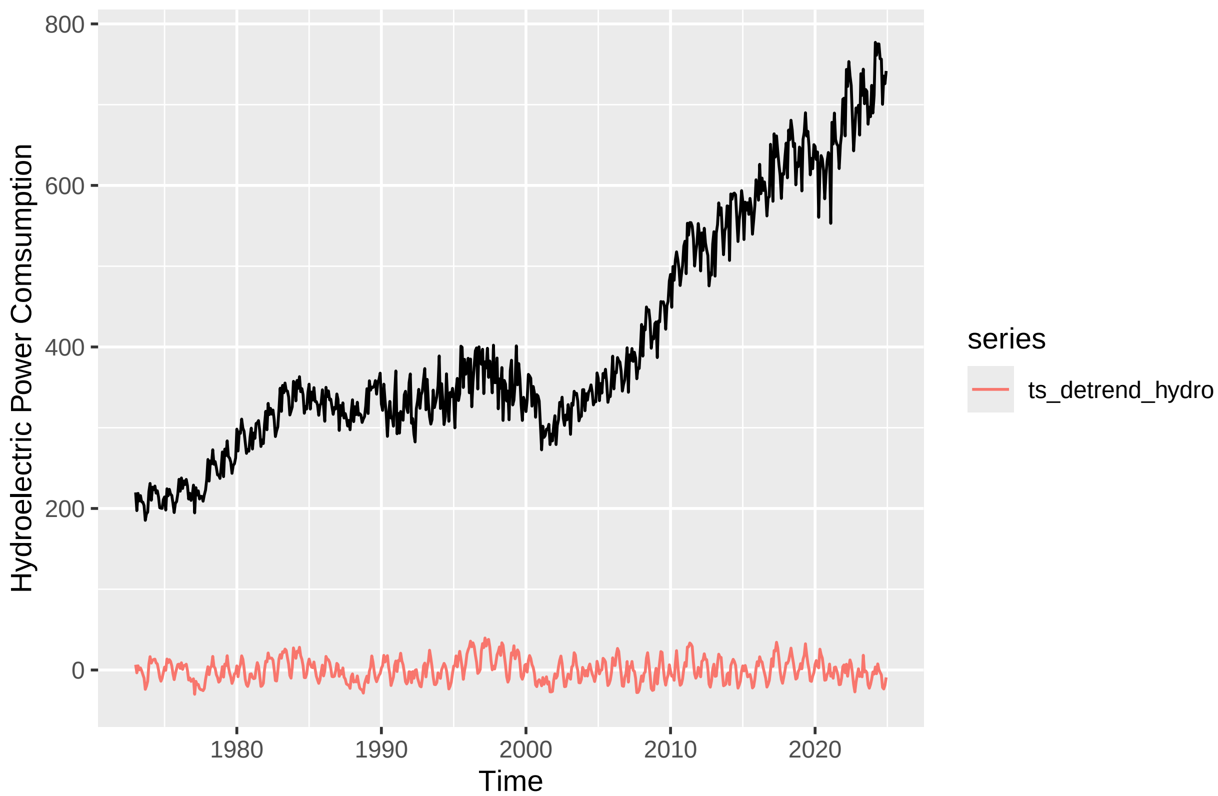
```r
#need to conv from numeric to time series

ts_detrend_tren <- ts(detrend_tren_data, frequency = 12, start = c(1973,1))
#monthly data (freq 12), starts at year 1973, month 1

ts_tren <- ts(shortened_df[,2], frequency = 12, start = c(1973,1))

autoplot(ts_tren)+
  autolayer(ts_detrend_tren)+
  labs(y = "Total Renewable Energy Production")
```




```
#Hydropower
#remove the trend from series total series
detrend_hydro_data <- shortened_df[[3]] - (beta1_hydro + beta2_hydro*t)
#subtracting the equation (with the stored coeffs and the t value)
class(detrend_hydro_data)
```

```
## [1] "numeric"
```

```
#need to conv from numeric to time series

ts_detrend_hydro <- ts(detrend_hydro_data, frequency = 12, start = c(1973,1))
#monthly data (freq 12), starts at year 1973, month 1

ts_hydro <- ts(shortened_df[,2], frequency = 12, start = c(1973,1))

autoplot(ts_hydro)+
  autolayer(ts_detrend_hydro)+
  labs(y = "Hydroelectric Power Comsumption")
```

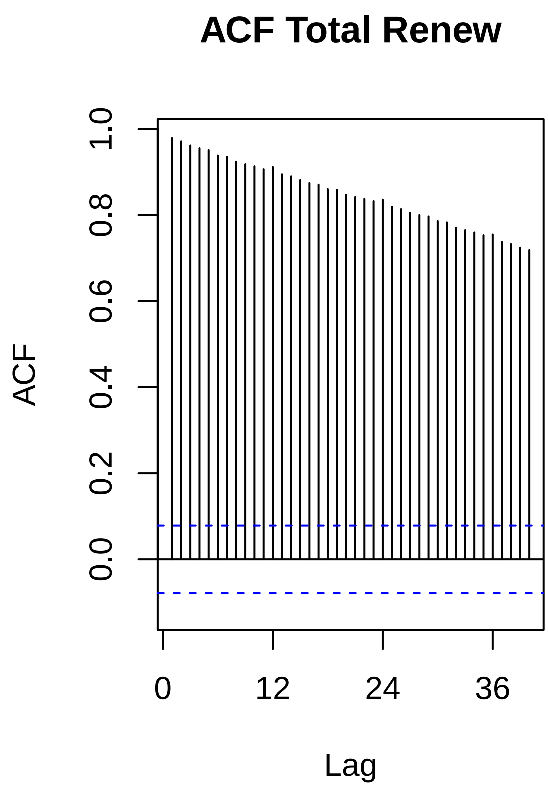


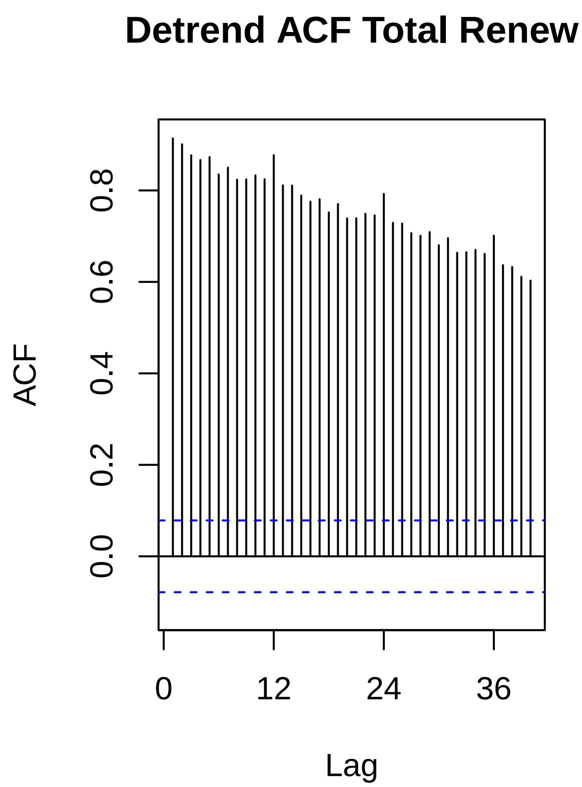
The level of the series for each series was dropped to be centered around zero and the increasing trend was removed.

Q5

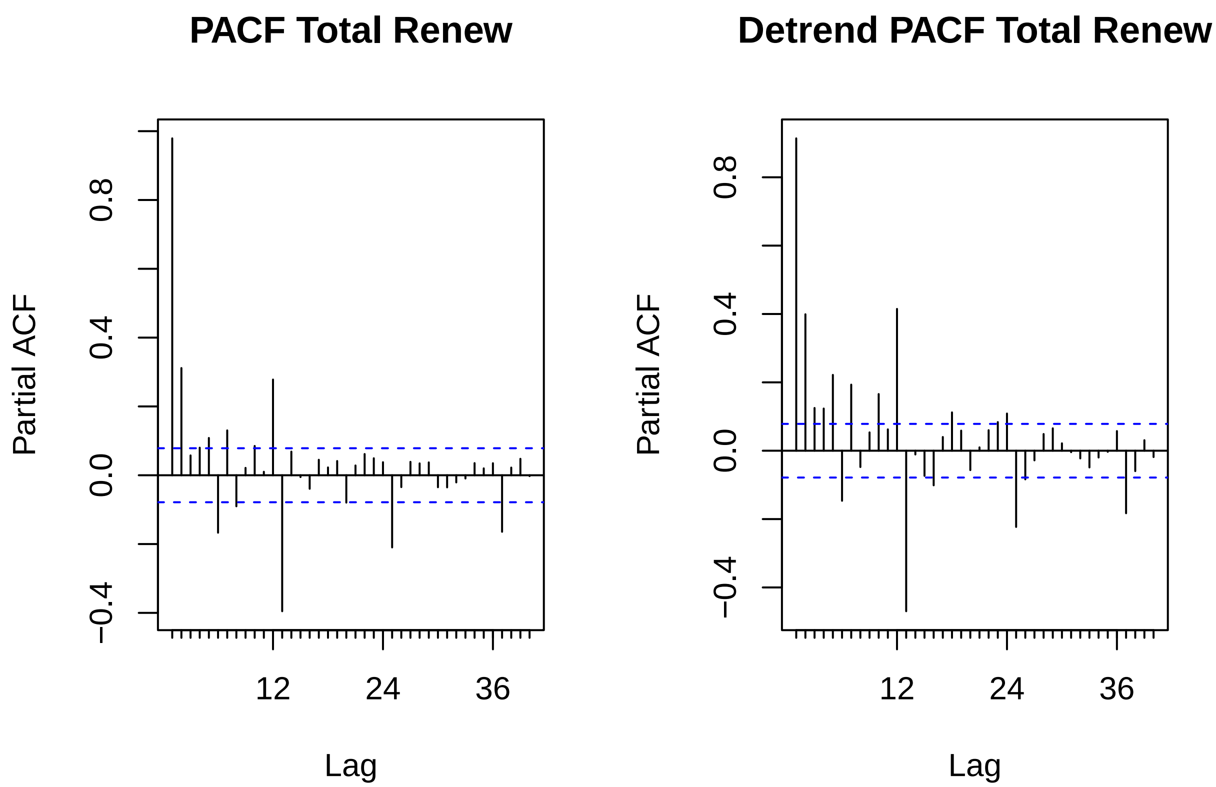
Plot ACF and PACF for the detrended series and compare with the plots from Q1. You may use plot_grid() again to get them side by side to make it easier to compare. Did the plots change? How?

```
#ACF Comp for Total Renewable
par(mfrow=c(1,2))  #place plot side by side
Acf(ts_energy_data[,1],lag.max=40,main=paste("ACF Total Renew",sep=""))
Acf(ts_detrend_tren,lag.max=40,main=paste("Detrend ACF Total Renew",sep=""))
```

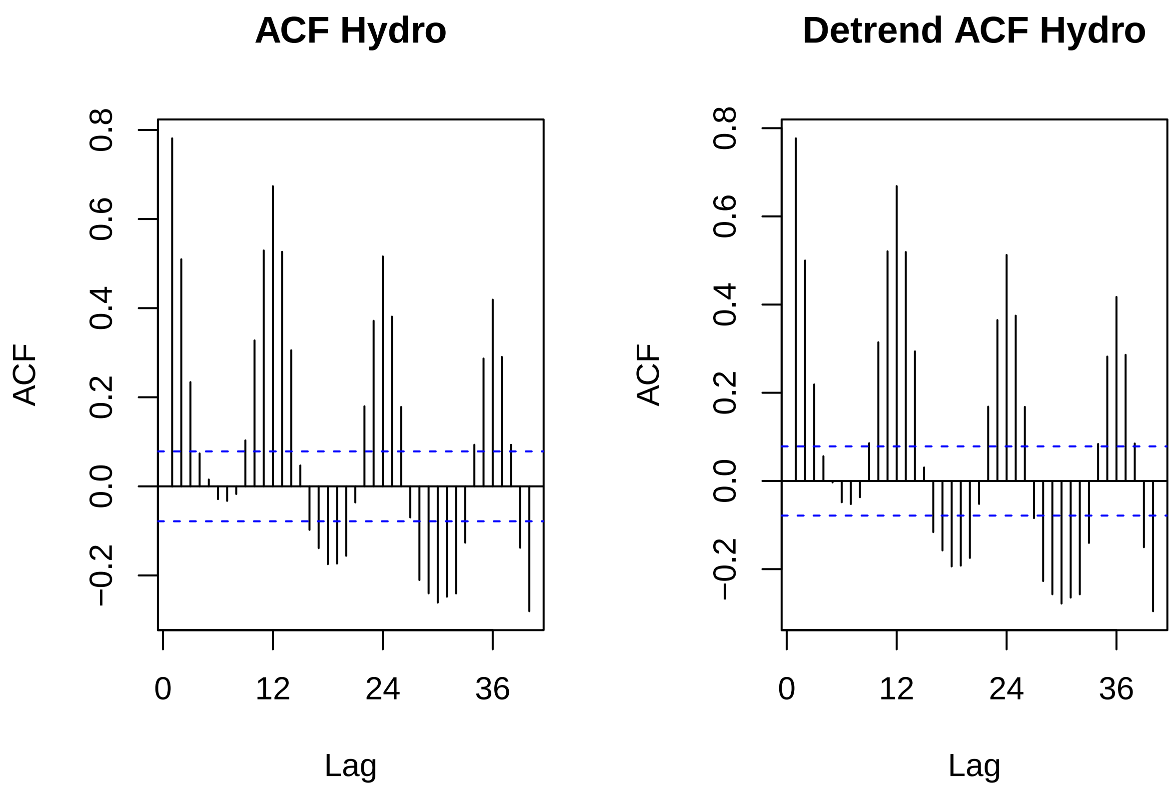




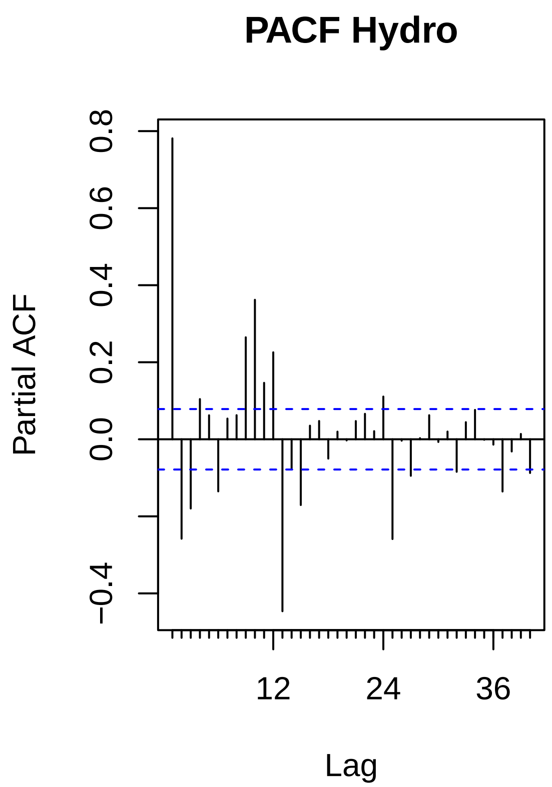

```r
#PACF Comp for Total Renewable
par(mfrow=c(1,2))  #place plot side by side
Pacf(ts_energy_data[,1],lag.max=40,main=paste("PACF Total Renew",sep=""))
Pacf(ts_detrend_tren,lag.max=40,main=paste("Detrend PACF Total Renew",sep=""))
```

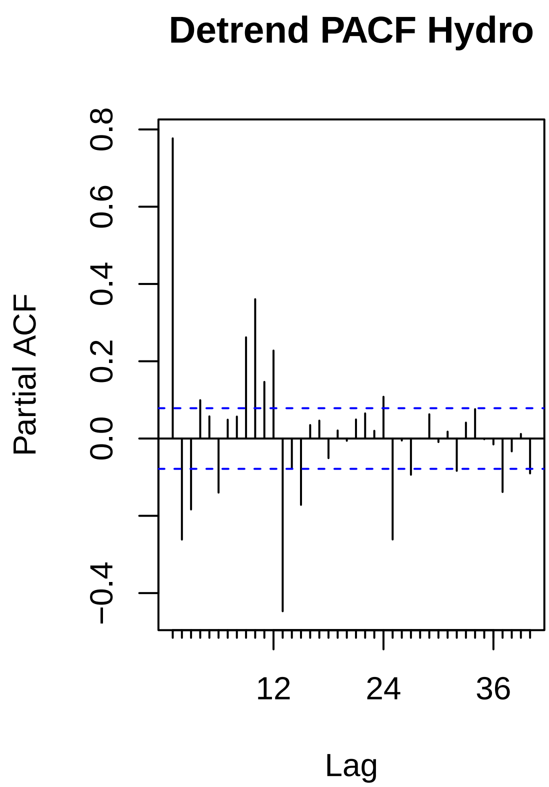



```
#ACF Comp for Hydro
par(mfrow=c(1,2))  #place plot side by side
Acf(ts_energy_data[,2],lag.max=40,main=paste("ACF Hydro",sep=""))
Acf(ts_detrend_hydro,lag.max=40,main=paste("Detrend ACF Hydro",sep=""))
```




```r
#PACF Comp for Honest
par(mfrow=c(1,2))  #place plot side by side
Pacf(ts_energy_data[,2],lag.max=40,main=paste("PACF Hydro",sep=""))
Pacf(ts_detrend_hydro,lag.max=40,main=paste("Detrend PACF Hydro",sep=""))
```





For the Total renewable energy ACF plots, a potential seasonal trend was emphasized in the detrended plot. However, ACF remained high with a slow decline, showing strong memory in the series. In the hydroelectric power consumption series, the original and detrended plots appear very similar, indicating that the seasonal component has the greatest influence on the series.

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, both series seem like they have a seasonal component due to the yearly oscillations present in their time series. The PACF plots for both series also oscillate around zero.

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
## Total Renewable Deseason Series ####################################
#dummies

## in assignment --> will remove trend and remove seasonality separately
dummies_tren <- seasonaldummy(ts_energy_data[,1])

#regress the detrended inflow data as a data frame on the dummies
seas_means_model_tren <- lm(ts_energy_data[,1]~dummies_tren)

#summary of the regression
summary(seas_means_model_tren)
```

```
##
## Call:
## lm(formula = ts_energy_data[, 1] ~ dummies_tren)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -206.22  -92.92  -56.72  118.76  367.32
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      417.265     20.372  20.482   <2e-16 ***
## dummies_trenJan   -4.287     28.811  -0.149    0.882
## dummies_trenFeb  -40.496     28.811  -1.406    0.160
## dummies_trenMar   -1.537     28.811  -0.053    0.957
## dummies_trenApr  -14.075     28.811  -0.489    0.625
## dummies_trenMay    1.083     28.811   0.038    0.970
## dummies_trenJun   -9.440     28.811  -0.328    0.743
## dummies_trenJul   -3.312     28.811  -0.115    0.909
## dummies_trenAug  -10.704     28.811  -0.372    0.710
## dummies_trenSep  -35.317     28.811  -1.226    0.221
## dummies_trenOct  -19.937     28.811  -0.692    0.489
## dummies_trenNov  -20.617     28.811  -0.716    0.474
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 146.9 on 612 degrees of freedom
## Multiple R-squared:  0.007996,   Adjusted R-squared:  -0.009834
## F-statistic: 0.4484 on 11 and 612 DF,  p-value: 0.9336
```

```r
## Hydro Deseason Series ####################################

#dummies

## in assignment --> will remove trend and remove seasonality separately
dummies_hydro <- seasonaldummy(ts_energy_data[,2])

#regress the detrended inflow data as a data frame on the dummies
seas_means_model_hydro <- lm(ts_energy_data[,2]~dummies_hydro)

#summary of the regression
summary(seas_means_model_hydro)
```

```
##
## Call:
## lm(formula = ts_energy_data[, 2] ~ dummies_hydro)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.181  -6.366  -0.465   6.293  32.360
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         79.724      1.434  55.595  < 2e-16 ***
## dummies_hydroJan     5.173      2.028   2.551 0.010986 *
## dummies_hydroFeb    -2.211      2.028  -1.090 0.276086
## dummies_hydroMar     7.313      2.028   3.606 0.000337 ***
## dummies_hydroApr     5.756      2.028   2.838 0.004684 **
## dummies_hydroMay    14.285      2.028   7.044 5.04e-12 ***
## dummies_hydroJun    11.066      2.028   5.457 7.05e-08 ***
## dummies_hydroJul     4.302      2.028   2.121 0.034285 *
## dummies_hydroAug    -5.200      2.028  -2.564 0.010583 *
## dummies_hydroSep   -16.389      2.028  -8.081 3.43e-15 ***
## dummies_hydroOct   -16.329      2.028  -8.052 4.27e-15 ***
## dummies_hydroNov   -10.782      2.028  -5.316 1.48e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.34 on 612 degrees of freedom
## Multiple R-squared:  0.4755, Adjusted R-squared:  0.4661
## F-statistic: 50.44 on 11 and 612 DF,  p-value: < 2.2e-16
```

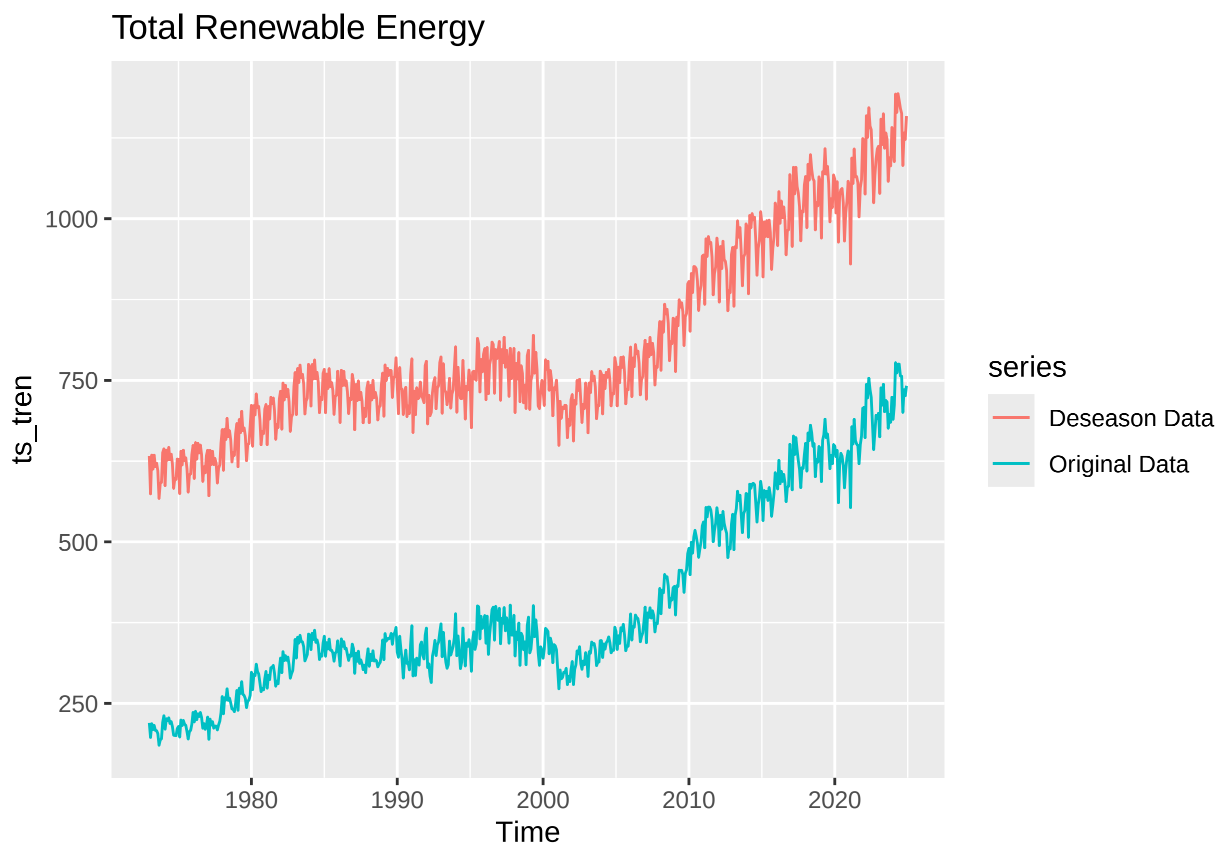
For the total renewable energy series, only December has a significant seasonal trend, indicating that seasonality is not a strong component of the behavior of the series and that seasonal trends might occur on an annual basis rather than monthly. For the hydroelectric power series, all months except for February have significant seasonal trends. In Q6, I said that both series would have seasonal trends. However, only the hydroelectric power series has a true 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?

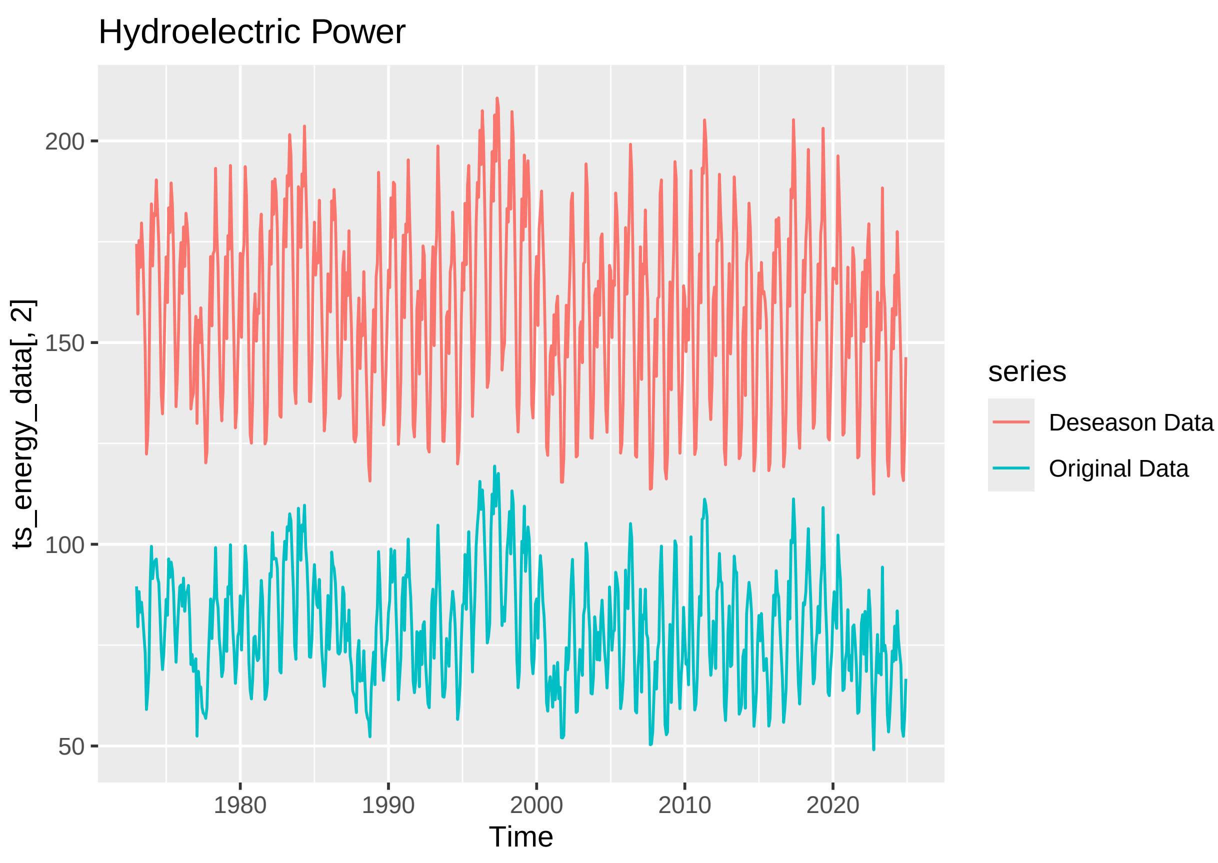
```
## Total Renewable Deseason Series ####################################

#storing the coefficients
beta_int_tren <- seas_means_model_tren$coefficients[1]
beta_coeff_tren <- seas_means_model_tren$coefficients[2:12]


season_inflow_data_tren <- array(0, nobs)

for (i in 1: nobs){
  season_inflow_data_tren[i] <- beta_int_tren+beta_coeff_tren %*%
    dummies_tren[i,]
}
```

```
deseason_data_tren <- energy_data[,2] + season_inflow_data_tren

ts_deseason_tren <- ts(deseason_data_tren,
                              start = c(1973,1),
                              frequency = 12)

ts_season_tren <- ts(season_inflow_data_tren,
                start = c(1973,1),
                frequency = 12)

#plots
autoplot(ts_tren, series = "Original Data")+
  autolayer(ts_deseason_tren, series = "Deseason Data")+
  labs(title = "Total Renewable Energy")
```




```
## Hydro Deseason Series ####################################
#storing the coefficients
beta_int_hydro <- seas_means_model_hydro$coefficients[1]
beta_coeff_hydro <- seas_means_model_hydro$coefficients[2:12]


season_inflow_data_hydro <- array(0, nobs)

for (i in 1: nobs){
```

```r
  season_inflow_data_hydro[i] <- beta_int_hydro+beta_coeff_hydro %*%
    dummies_hydro[i,]
}


deseason_data_hydro <- energy_data[,3] + season_inflow_data_hydro

ts_deseason_hydro <- ts(deseason_data_hydro,
                              start = c(1973,1),
                              frequency = 12)

ts_season_hydro <- ts(season_inflow_data_hydro,
                start = c(1973,1),
                frequency = 12)

#plots
autoplot(ts_energy_data[,2], series = "Original Data")+
  autolayer(ts_deseason_hydro, series = "Deseason Data")+
  labs(title = "Hydroelectric Power")
```

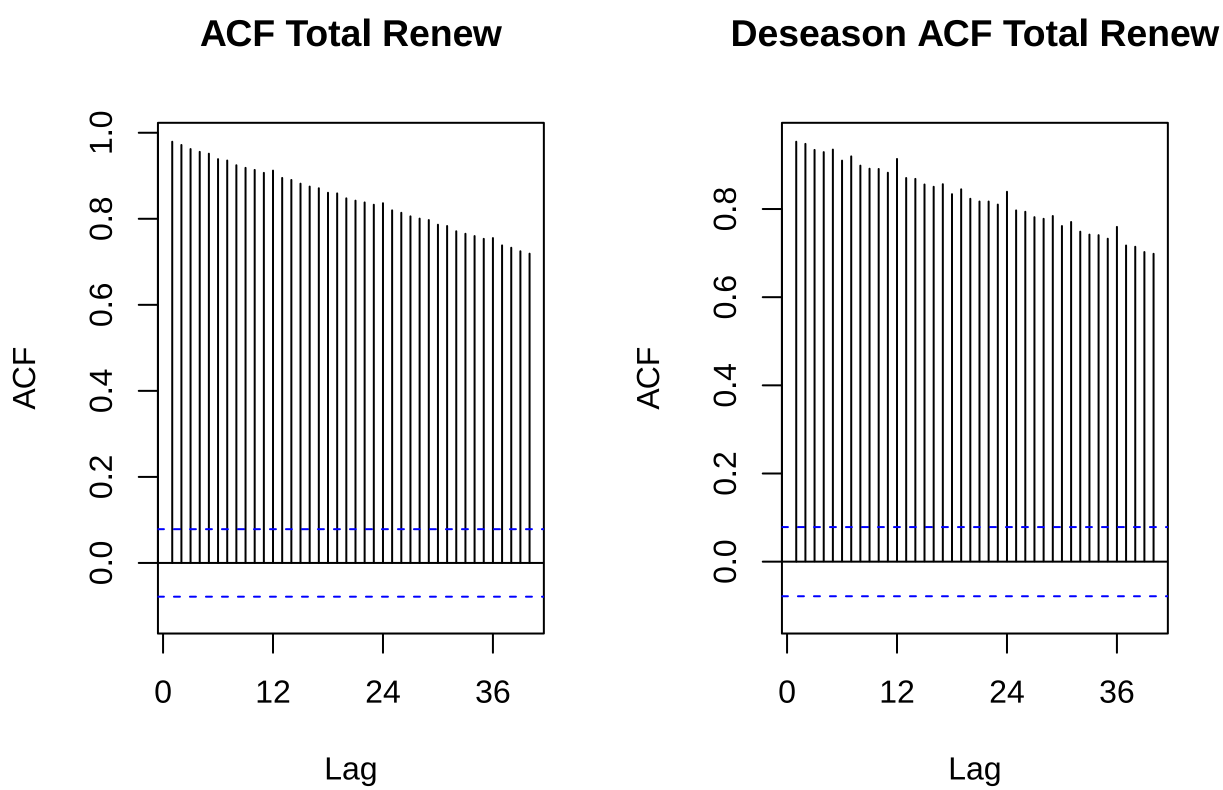


For both series, when comparing the deseasoned series with the original series, both the level and range of the oscillations decreased in the deseasoned series.

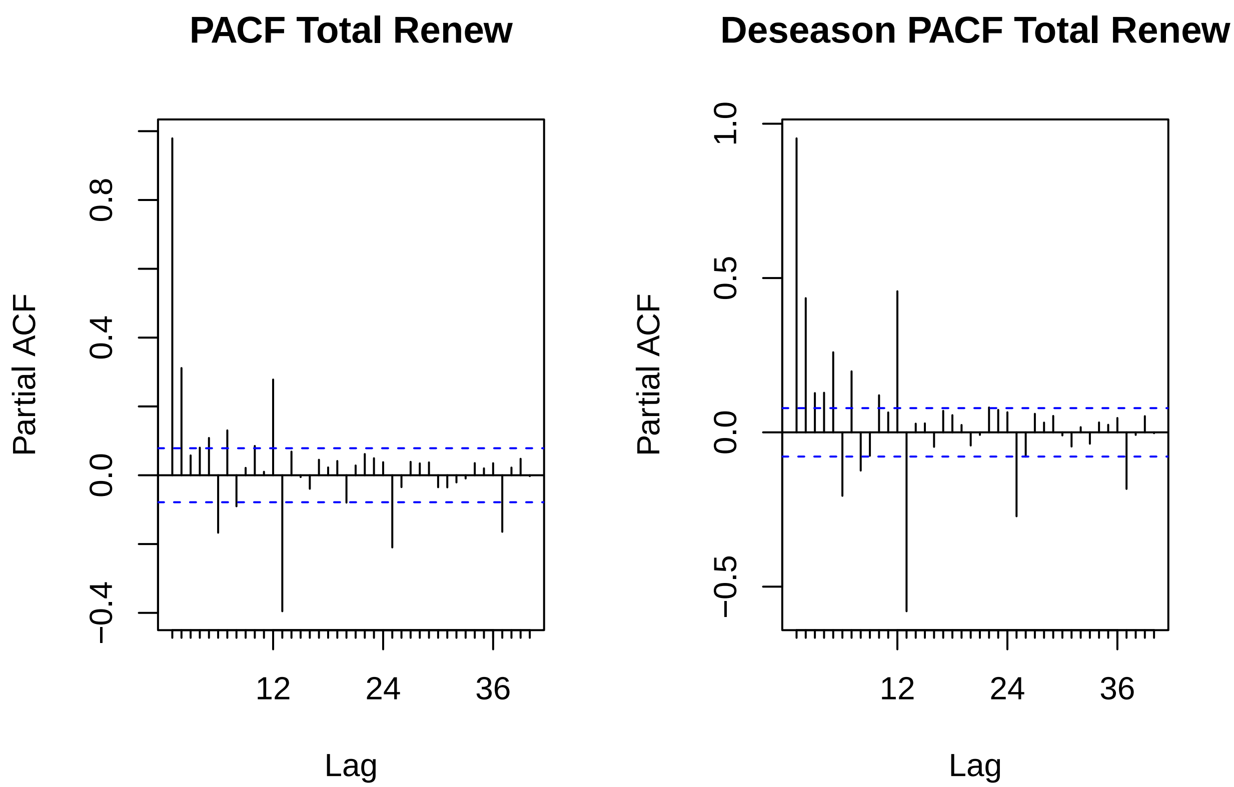
Q9

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?

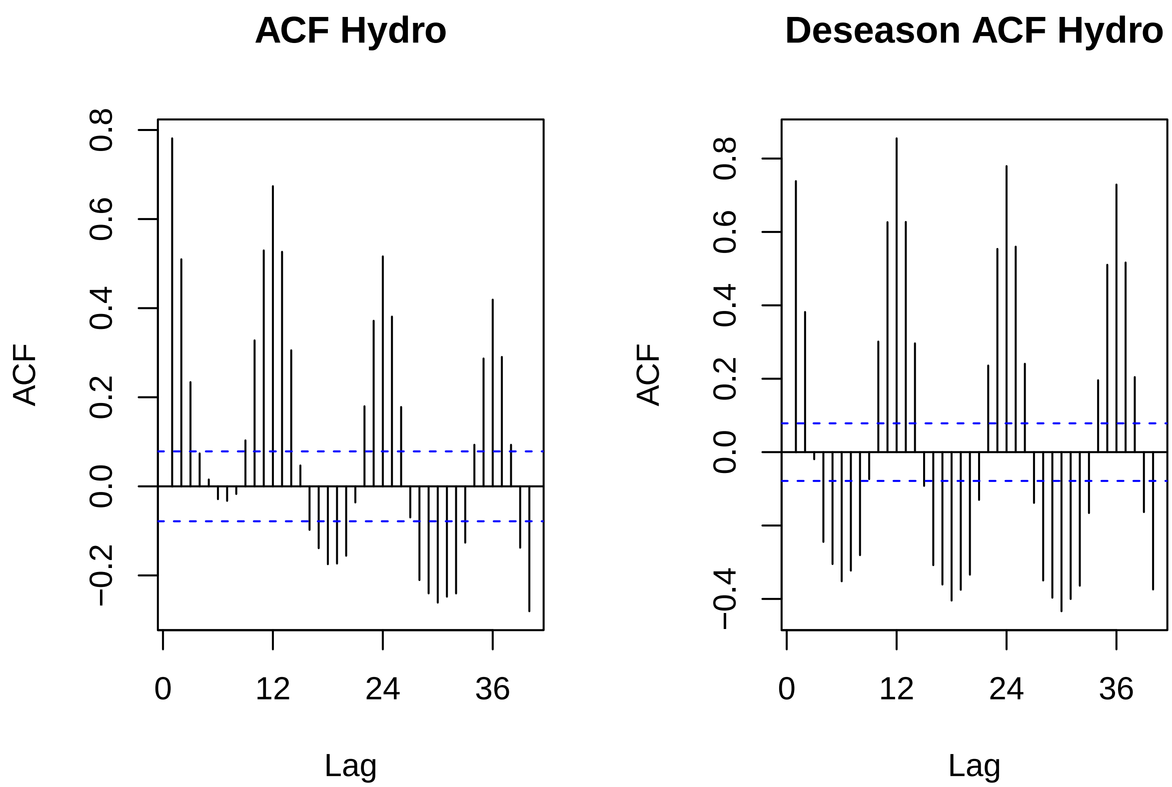
```
#ACF Comp for Total Renewable
par(mfrow=c(1,2))  #place plot side by side
Acf(ts_energy_data[,1],lag.max=40,main=paste("ACF Total Renew",sep=""))
Acf(ts_deseason_tren,lag.max=40,main=paste("Deseason ACF Total Renew",sep=""))
```



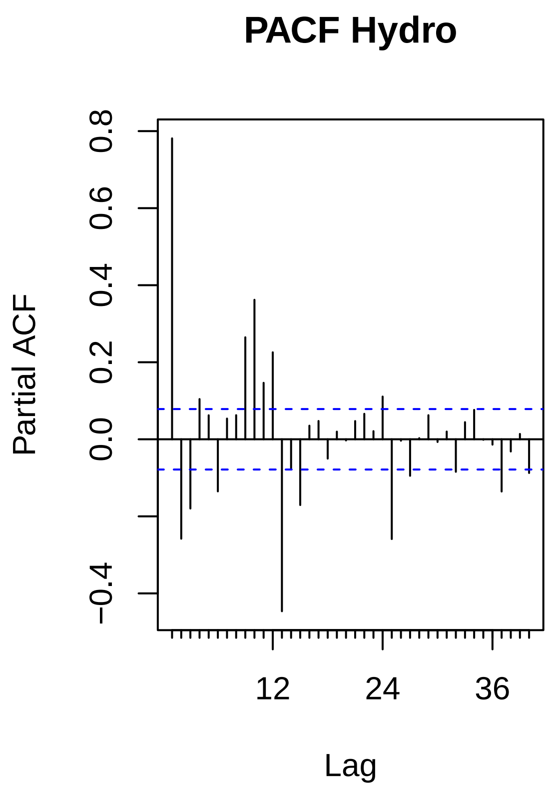

```
#PACF Comp for Total Renewable
par(mfrow=c(1,2))  #place plot side by side
Pacf(ts_energy_data[,1],lag.max=40,main=paste("PACF Total Renew",sep=""))
Pacf(ts_deseason_tren,lag.max=40,main=paste("Deseason PACF Total Renew",sep=""))
```

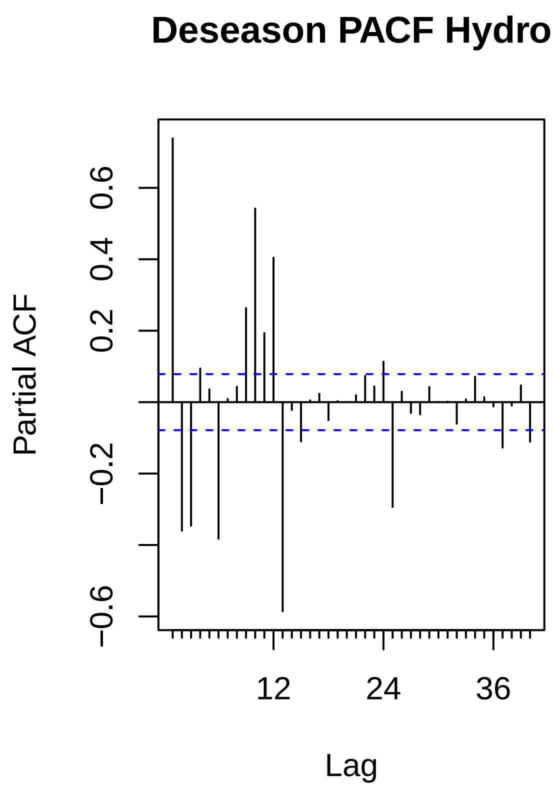



```r
#ACF Comp for Hydro
par(mfrow=c(1,2))  #place plot side by side
Acf(ts_energy_data[,2],lag.max=40,main=paste("ACF Hydro",sep=""))
Acf(ts_deseason_hydro,lag.max=40,main=paste("Deseason ACF Hydro",sep=""))
```




```
#PACF Comp for Honest
par(mfrow=c(1,2))  #place plot side by side
Pacf(ts_energy_data[,2],lag.max=40,main=paste("PACF Hydro",sep=""))
Pacf(ts_deseason_hydro,lag.max=40,main=paste("Deseason PACF Hydro",sep=""))
```





The ACF and PACF plots did not seem to significantly change between the original and deseasoned total renewable energy plots. However, removing the seasonal trend in both the total renewable energy and hydroelectric power PACF plots allows us to visualize the short-term lags having stronger correlations, indicating that removing seasonality helped reveal the underlying non-seasonal behavior of the series.