ENV 790.30 - Time Series Analysis for Energy Data | Spring 2024

Assignment 3 - Due date 02/01/24

VincientWhatley

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_A02_Sp24.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 2022 **Monthly** Energy Review. Once again you will work only with the following columns: Total Biomass Energy Production, Total Renewable Energy Production, Hydroelectric Power Consumption. Create a data frame structure with these three 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(Kendall)
library(tseries)
library(ggplot2)
library(cowplot)
library(gridExtra)
```

Data Manipulation

```
# reading in data with read.csv function
energy_data_raw <- read.csv(file="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Sour

#Triming data for specified colmunis
energydata <- energy_data_raw[,1:6]

#Removing unnecessary columns, renaming columns
energydata <- energydata[, -c(2, 3)]
colnames(energydata)=c("Month", "TBEP", "TREP","HEPC")

#convert month day year

mdy_date <- paste(energydata[,1]," 01",sep="")
mdy_date1 <- as.Date(mdy_date, format = "%Y %B %d")  #function my from package lubridate
head(mdy_date1)
```

```
## [1] "1973-01-01" "1973-02-01" "1973-03-01" "1973-04-01" "1973-05-01"
## [6] "1973-06-01"
```

```
#add that to energydata
energydata <- cbind(mdy_date1,energydata[,2:4]) #cbind stands for column bind
head(energydata)
```

```
##    mdy_date1    TBEP    TREP   HEPC
## 1 1973-01-01 129.787 219.839 89.562
## 2 1973-02-01 117.338 197.330 79.544
## 3 1973-03-01 129.938 218.686 88.284
## 4 1973-04-01 125.636 209.330 83.152
## 5 1973-05-01 129.834 215.982 85.643
## 6 1973-06-01 125.611 208.249 82.060
```

```
#Transform data
ts_TBEP_data <- ts(energydata [,"TBEP"], start= c(1973,1), frequency = 12)
ts_TREP_data <- ts(energydata [,"TREP"], start= c(1973,1), frequency = 12)
ts_HEPC_data <- ts(energydata [,"HEPC"], start= c(1973,1), frequency = 12)
```

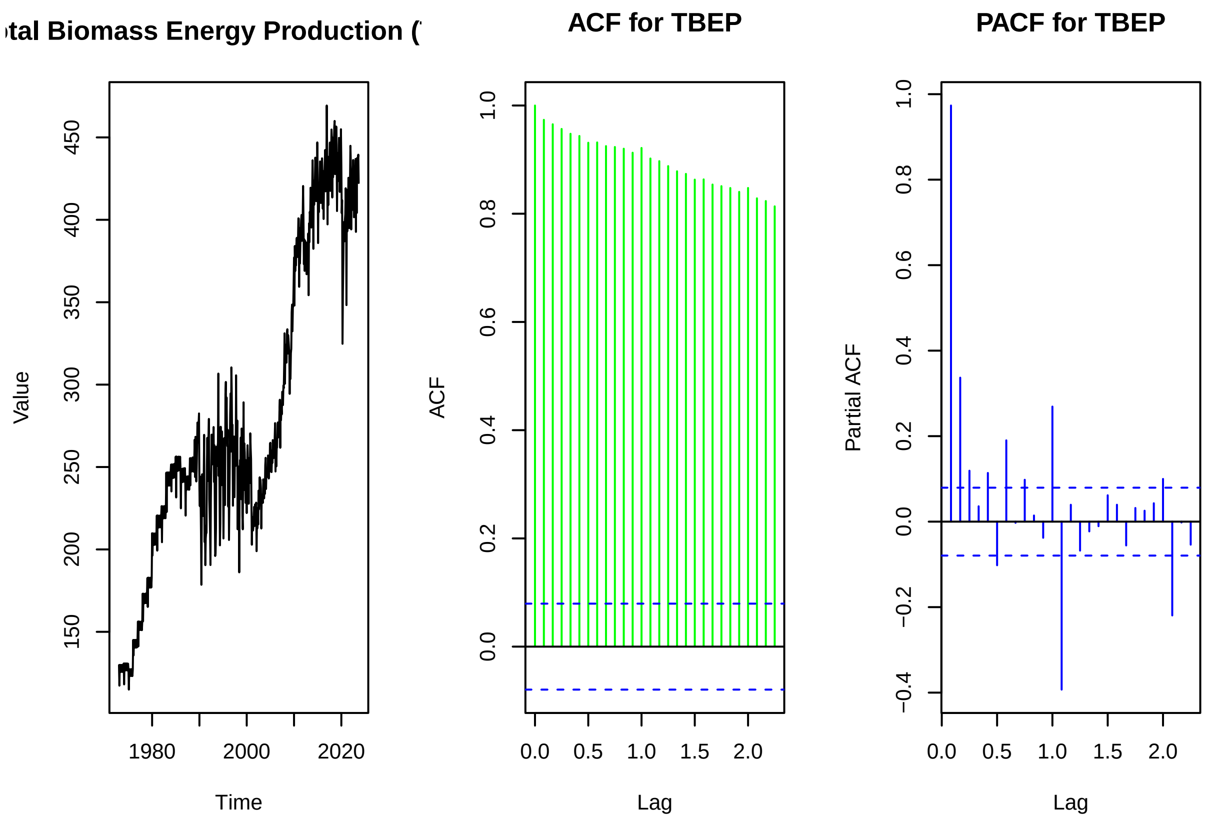
##Trend Component

Q1

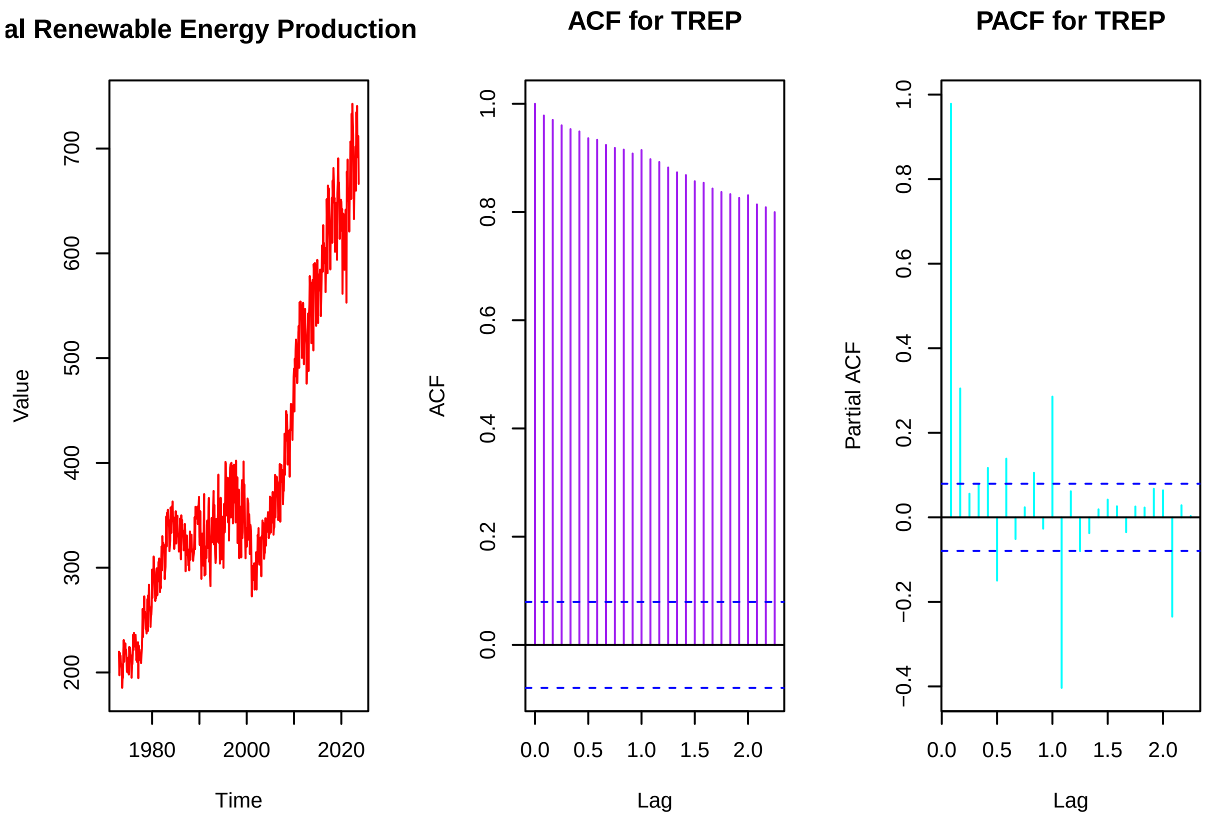
For each time series, i.e., Renewable Energy Production and Hydroelectric Consumption create three plots: one with time series, one with the ACF and with the PACF. You may use the some code form A2, but I want all the three plots side by side as in a grid. (Hint: use function `plot_grid()` from the `cowplot` package)

```
#The par mfrow function allows us to create the grid for our plots.
#This is a 1 row but 3 collumn grid.
par(mfrow=c(1,3))
```

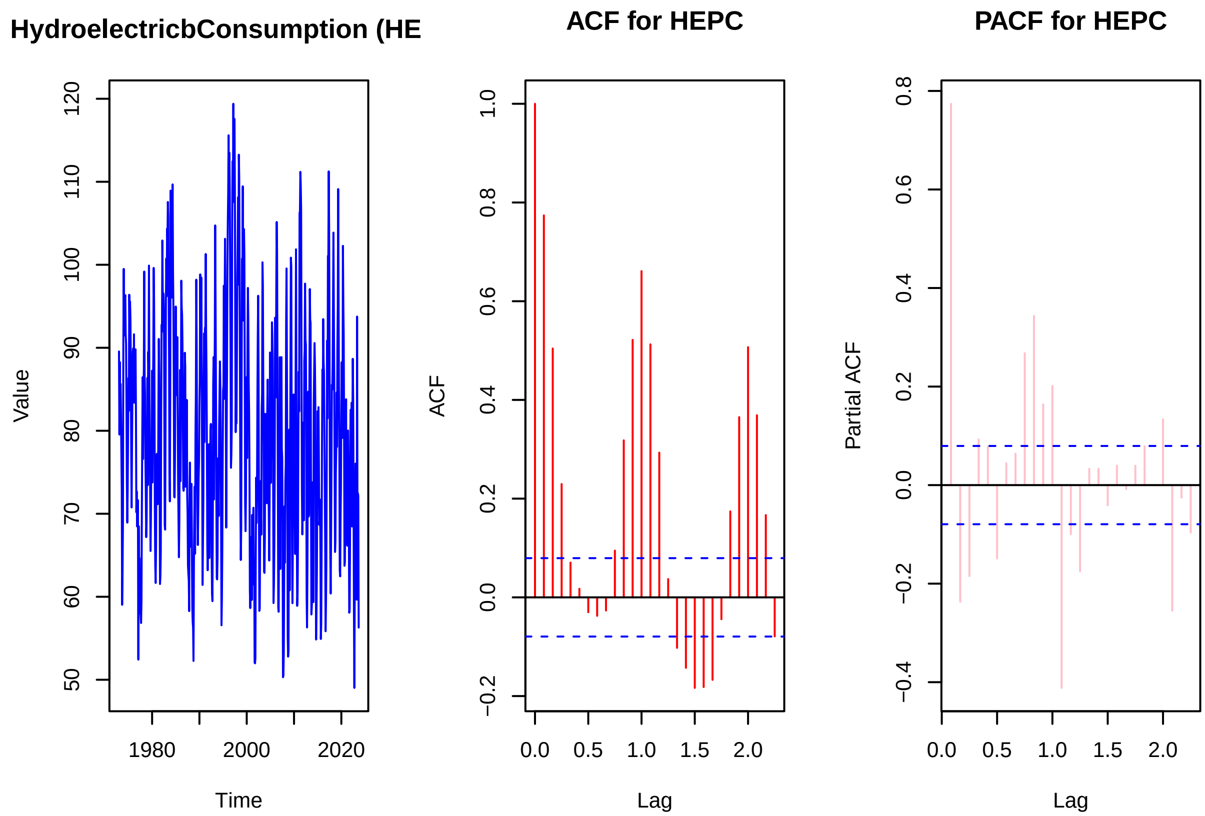
```
#Plots for TBEP
plot(ts_TBEP_data, main="Total Biomass Energy Production (TBEP)",ylab="Value",col="black")
  acf(ts_TBEP_data, main="ACF for TBEP",col="green")
    pacf(ts_TBEP_data, main="PACF for TBEP",col="blue")
```




```
#Plot for TREP
plot(ts_TREP_data, main="Total Renewable Energy Production (TREP)", ylab="Value", col="red")
  acf(ts_TREP_data, main="ACF for TREP", col="purple")
    pacf(ts_TREP_data, main="PACF for TREP", col="cyan")
```




```r
#Plot for HEPC
plot(ts_HEPC_data, main="HydroelectricbConsumption (HEPC)",ylab="Value",col="blue")
  acf(ts_HEPC_data, main="ACF for HEPC",col="red")
    pacf(ts_HEPC_data, main = "PACF for HEPC",col="pink")
```




```r
#This is to reset the layout for any future plots.
par(mfrow=c(1,1))
```

Q2

From the plot in Q1, do the series Total Biomass Energy Production, Total Renewable Energy Production, Hydroelectric Power Consumption appear to have a trend? If yes, what kind of trend?

Both TREP and TBEP demonstrate the a clear exponential positive trend and their ACF and PACF demonstrate a linear trend. As for the HEPC, it was kind of hard to identify some sort of trend with the time series plot but it becomes cyclicality.

Q3

Use the *lm()* function to fit a linear trend to the three 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.

```r
# Fit linear trend to TBEP time series

lm_TBEP <-lm(ts_TBEP_data~time(ts_TBEP_data))
summary(lm_TBEP)
```

```
##
```

```
## Call:
## lm(formula = ts_TBEP_data ~ time(ts_TBEP_data))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -102.344  -23.754    5.491   31.980   83.154
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -1.116e+04  2.170e+02  -51.43   <2e-16 ***
## time(ts_TBEP_data)  5.726e+00  1.086e-01   52.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.26 on 607 degrees of freedom
## Multiple R-squared:  0.8208, Adjusted R-squared:  0.8205
## F-statistic:  2780 on 1 and 607 DF,  p-value: < 2.2e-16
```

```r
beta0_TBEP=as.numeric(lm_TBEP$coefficients[1])  #first coefficient is the intercept term or beta0
beta1_TBEP=as.numeric(lm_TBEP$coefficients[2])  #second coefficient is the slope or beta1
# Fit linear trend to TREP time series

lm_TREP <- lm(ts_TREP_data~time(ts_TREP_data))
summary(lm_TREP)
```

```
##
## Call:
## lm(formula = ts_TREP_data ~ time(ts_TREP_data))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -148.27  -35.63   11.58   41.51  144.27
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -1.649e+04  3.339e+02  -49.38   <2e-16 ***
## time(ts_TREP_data)  8.448e+00  1.671e-01   50.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.41 on 607 degrees of freedom
## Multiple R-squared:  0.8081, Adjusted R-squared:  0.8078
## F-statistic:  2557 on 1 and 607 DF,  p-value: < 2.2e-16
```

```r
beta0_TREP=as.numeric(lm_TREP$coefficients[1])  #first coefficient is the intercept term or beta0
beta1_TREP=as.numeric(lm_TREP$coefficients[2])  #second coefficient is the slope or beta1

# Fit linear trend to HEPC time serie

lm_HEPC <- lm(ts_HEPC_data~time(ts_HEPC_data))
summary(lm_HEPC)
```

```
##
```

```
## Call:
## lm(formula = ts_HEPC_data ~ time(ts_HEPC_data))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.818 -10.620  -0.669   9.357  39.528
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         315.91687   77.67397   4.067 5.39e-05 ***
## time(ts_HEPC_data)   -0.11819    0.03887  -3.041  0.00246 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.05 on 607 degrees of freedom
## Multiple R-squared:  0.015,  Adjusted R-squared:  0.01338
## F-statistic: 9.247 on 1 and 607 DF,  p-value: 0.002461
```

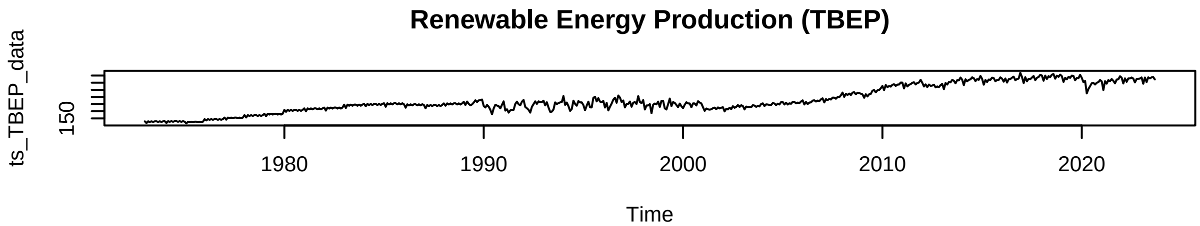
```
beta0_HEPC=as.numeric(lm_HEPC$coefficients[1])  #first coefficient is the intercept term or beta0
beta1_HEPC=as.numeric(lm_HEPC$coefficients[2])  #second coefficient is the slope or beta1
```

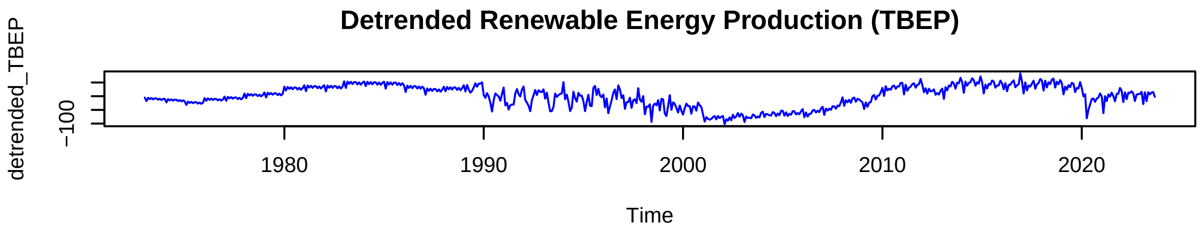
Q4

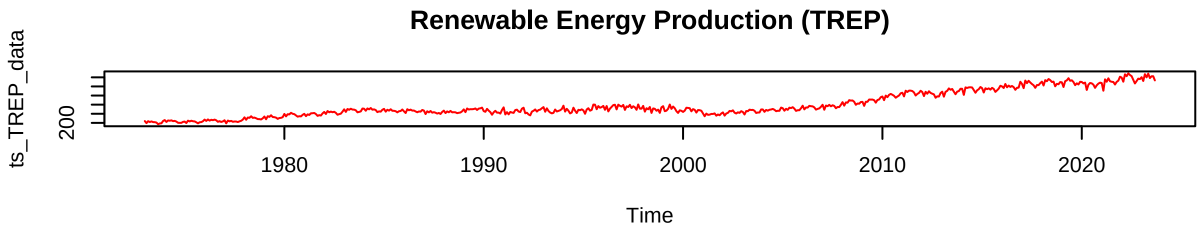
Use the regression coefficients from Q3 to detrend the series. Plot the detrended series and compare with the plots from Q1. What happened? Did anything change?

For the HEPC plots nothing really changed in the plots, they relatively similar. As for TBEP from 1990 to 2000 and 2010 to 2020 the seasonality becomes more prominent. For TREP data it the detrended data become more defined and flatter.

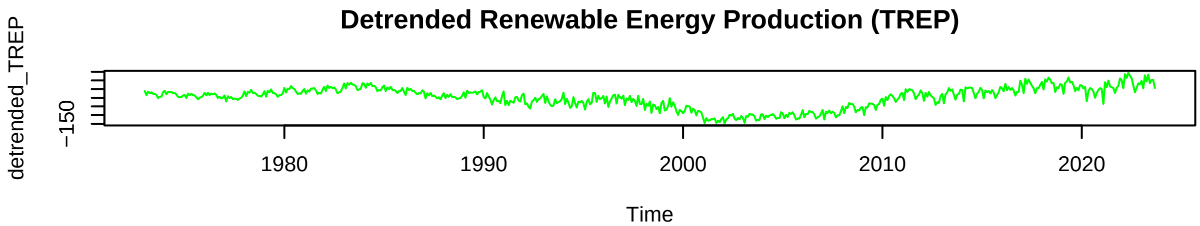
```
#Detrend TBEP time series
detrended_TBEP <- ts_TBEP_data-(beta0_TBEP + beta1_TBEP * time(ts_TBEP_data))

#Detrend TREP time series
detrended_TREP <- ts_TREP_data-(beta0_TREP + beta1_TREP * time(ts_TREP_data))

#Detrend HEPC time series
detrended_HEPC <- ts_HEPC_data-(beta0_HEPC + beta1_HEPC * time(ts_HEPC_data))

#Plot the detrended series
par(mfrow = c(3, 1))

#Detrended TBEP and original comparison
plot(ts_TBEP_data, main="Renewable Energy Production (TBEP)",ylab="ts_TBEP_data",col="black")
plot(detrended_TBEP, main = "Detrended Renewable Energy Production (TBEP)", ylab = "detrended_TBEP", col


#Detrended TREP
plot(ts_TREP_data, main="Renewable Energy Production (TREP)", ylab="ts_TREP_data", col="red")
```

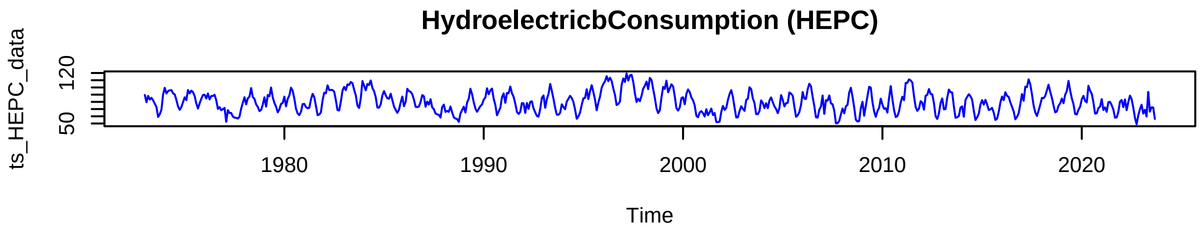


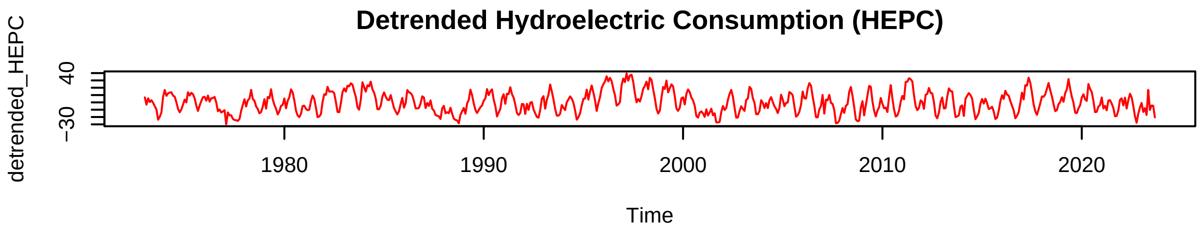





```
plot(detrended_TREP, main = "Detrended Renewable Energy Production (TREP)", ylab = "detrended_TREP", col

#Detrended HEPC
plot(ts_HEPC_data, main="HydroelectricbConsumption (HEPC)",ylab="ts_HEPC_data",col="blue")
plot(detrended_HEPC, main = "Detrended Hydroelectric Consumption (HEPC)", ylab = "detrended_HEPC", col =
```







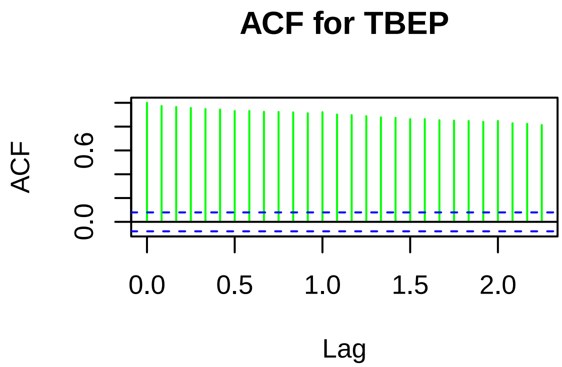

```
# Reset the layout
par(mfrow = c(1, 1))
```

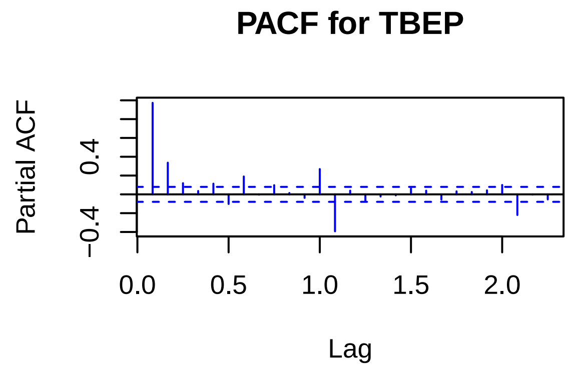
Q5

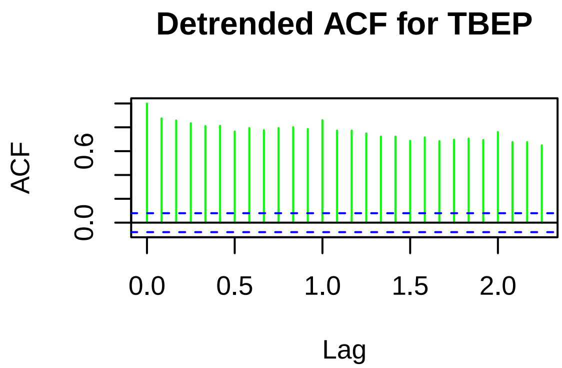
Plot ACF and PACF for the detrended series and compare with the plots from Q1. You may use plot_grid() again to get them side by side. nut mot mandatory. Did the plots change? How?

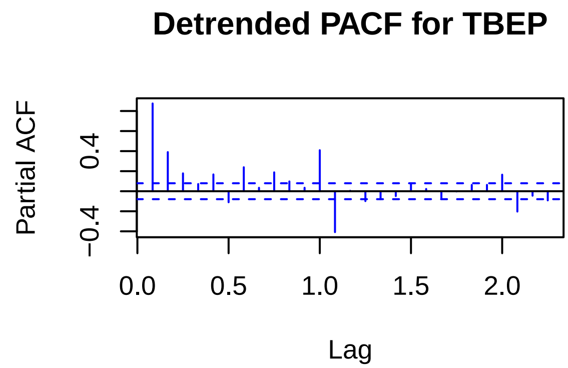
We see a similar story as the plots before in the previous questions. TREP and TBEP, show slight decay in for the ACF and PACF for detrended data. HEPC shows the same cyclicality for the both datasets.

```
#The par mfrow function allows us to create the grid for our plots.
#This is a 1 row but 3 collumn grid.
par(mfrow=c(2,2))

#Plots for TBEP
acf(ts_TBEP_data, main="ACF for TBEP",col="green")
pacf(ts_TBEP_data, main="PACF for TBEP",col="blue")
  acf(detrended_TBEP, main="Detrended ACF for TBEP",col="green")
  pacf(detrended_TBEP, main="Detrended PACF for TBEP",col="blue")
```

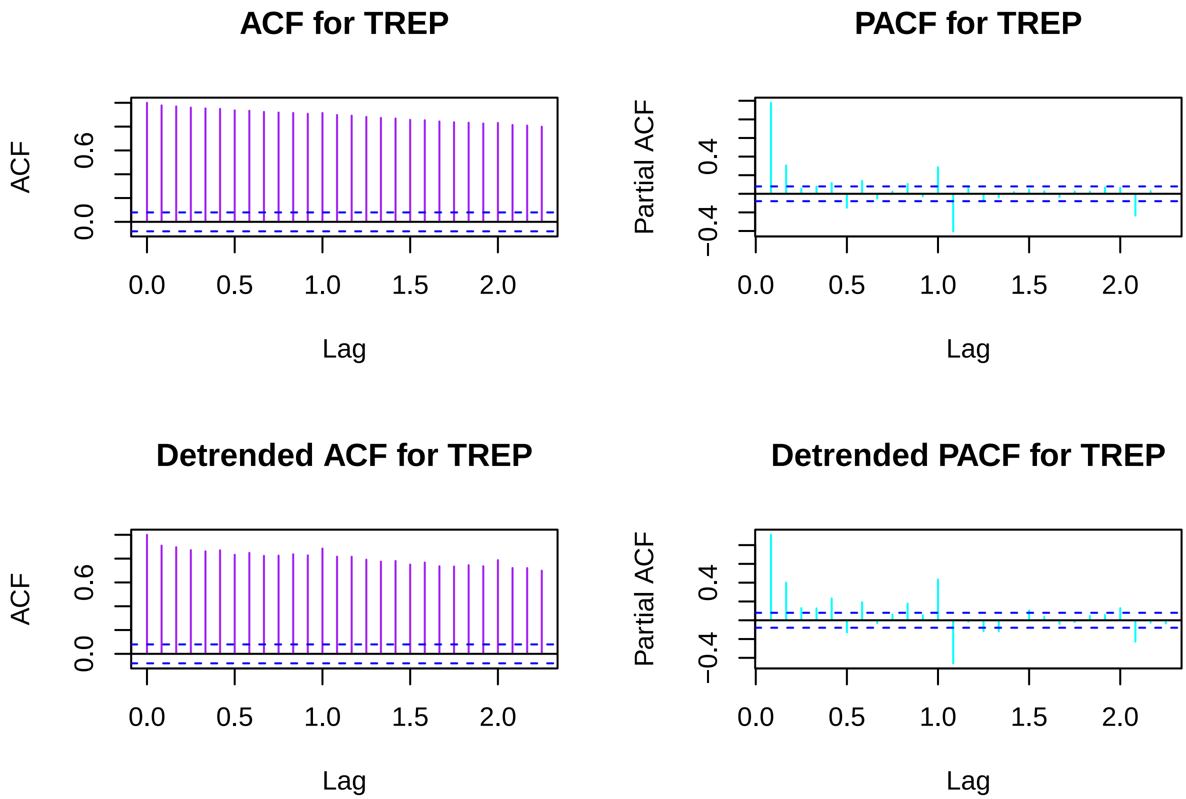




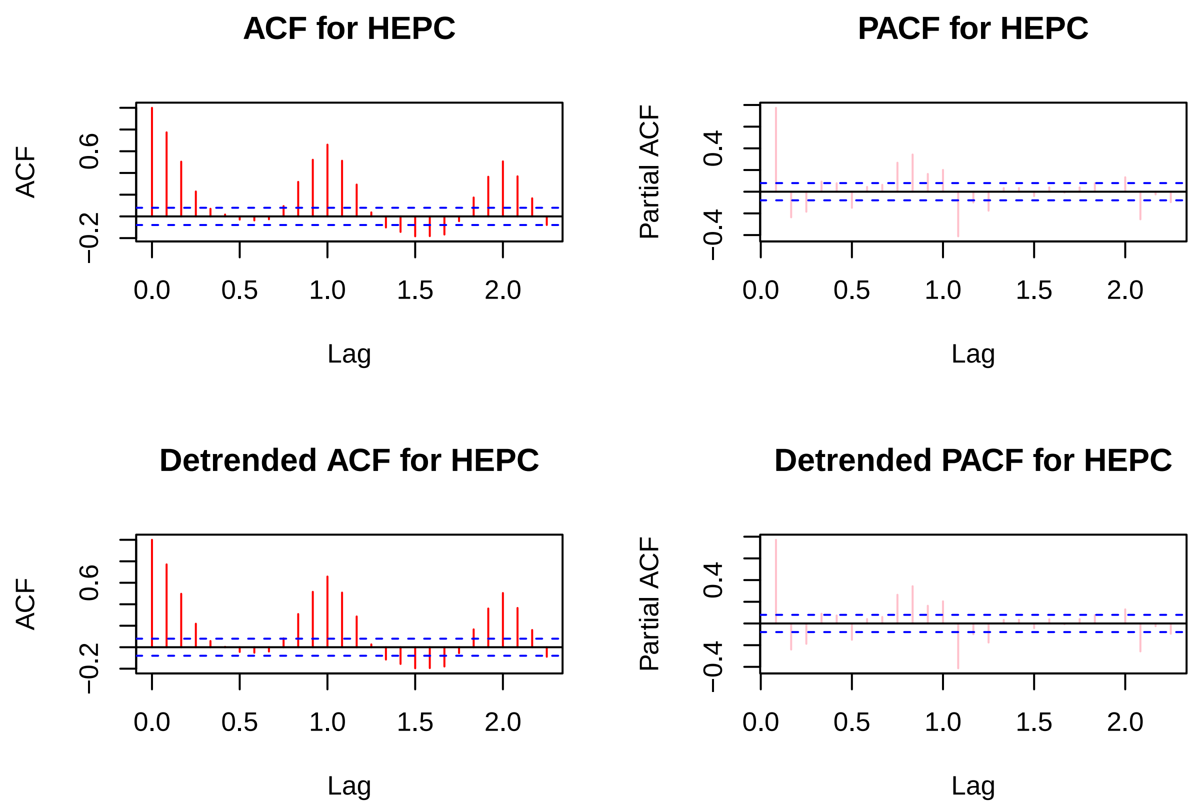





```r
#Plot for TREP
acf(ts_TREP_data, main="ACF for TREP", col="purple")
pacf(ts_TREP_data, main="PACF for TREP", col="cyan")
  acf(detrended_TREP, main="Detrended ACF for TREP", col="purple")
  pacf(detrended_TREP, main="Detrended PACF for TREP", col="cyan")
```




```r
#Plot for HEPC
acf(ts_HEPC_data, main="ACF for HEPC",col="red")
pacf(ts_HEPC_data, main = "PACF for HEPC",col="pink")
  acf(detrended_HEPC, main="Detrended ACF for HEPC",col="red")
  pacf(detrended_HEPC, main = "Detrended PACF for HEPC",col="pink")
```




```r
#This is to reset the layout for any future plots.
par(mfrow=c(1,1))
```

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.

Yes, the TREP and TBEP data show a scalloped the peak at each 1.0 lag and 2.0 while the HEPC demonstrates the same trend for both the regular data and detrended data.

Q7

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

Both TREP and TBEP show a significat result while evaluating the r-squared level are both above .81. But this is not the same for HEPC data which is only at .4841. This is diffrent from what I saw in question 6.

```r
#asfactor funtion allows us to inlcude the dummy variables within the seasonal...
#...means functions. The cycle function allows for us to get the cycle..
#...positions in the cycle of each obs, kind of like a looping function.


# Fit seasonal means model for TBEP
lm_seasonal_TBEP <- lm(ts_TBEP_data~time(ts_TBEP_data)+as.factor(cycle(ts_TBEP_data)))
beta_int_TBEP <- lm_seasonal_TBEP$coefficients[1]
beta_coeff_TBEP <- lm_seasonal_TBEP$coefficients[2:13]
summary(lm_seasonal_TBEP)
```

```
##
## Call:
## lm(formula = ts_TBEP_data ~ time(ts_TBEP_data) + as.factor(cycle(ts_TBEP_data)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -88.469 -24.460   4.517  32.489  72.990
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -1.115e+04  2.131e+02 -52.315  < 2e-16 ***
## time(ts_TBEP_data)                5.724e+00  1.066e-01  53.687  < 2e-16 ***
## as.factor(cycle(ts_TBEP_data))2  -3.009e+01  7.633e+00  -3.942 9.05e-05 ***
## as.factor(cycle(ts_TBEP_data))3  -8.132e+00  7.633e+00  -1.065  0.28719
## as.factor(cycle(ts_TBEP_data))4  -2.136e+01  7.633e+00  -2.798  0.00531 **
## as.factor(cycle(ts_TBEP_data))5  -1.416e+01  7.633e+00  -1.856  0.06400 .
## as.factor(cycle(ts_TBEP_data))6  -2.003e+01  7.633e+00  -2.624  0.00891 **
## as.factor(cycle(ts_TBEP_data))7  -4.755e+00  7.634e+00  -0.623  0.53359
## as.factor(cycle(ts_TBEP_data))8  -2.160e+00  7.634e+00  -0.283  0.77733
## as.factor(cycle(ts_TBEP_data))9  -1.532e+01  7.634e+00  -2.007  0.04522 *
## as.factor(cycle(ts_TBEP_data))10 -3.769e+00  7.671e+00  -0.491  0.62338
## as.factor(cycle(ts_TBEP_data))11 -9.896e+00  7.671e+00  -1.290  0.19757
## as.factor(cycle(ts_TBEP_data))12 -7.104e-01  7.672e+00  -0.093  0.92625
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.55 on 596 degrees of freedom
## Multiple R-squared:  0.8304, Adjusted R-squared:  0.827
## F-statistic: 243.1 on 12 and 596 DF,  p-value: < 2.2e-16
```

```r
# Fit seasonal means model for TREP
lm_seasonal_TREP <- lm(ts_TREP_data~time(ts_TREP_data)+as.factor(cycle(ts_TREP_data)))
beta_int_TREP <- lm_seasonal_TREP$coefficients[1]
beta_coeff_TREP <- lm_seasonal_TREP$coefficients[2:13]
summary(lm_seasonal_TREP)
```

```
##
## Call:
## lm(formula = ts_TREP_data ~ time(ts_TREP_data) + as.factor(cycle(ts_TREP_data)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -146.19  -37.88   14.21   42.20  128.77
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -1.648e+04  3.285e+02 -50.152  < 2e-16 ***
## time(ts_TREP_data)             8.450e+00  1.644e-01  51.409  < 2e-16 ***
## as.factor(cycle(ts_TREP_data))2  -3.814e+01  1.177e+01  -3.241  0.00126 **
## as.factor(cycle(ts_TREP_data))3  -4.416e-01  1.177e+01  -0.038  0.97008
## as.factor(cycle(ts_TREP_data))4  -1.377e+01  1.177e+01  -1.170  0.24241
## as.factor(cycle(ts_TREP_data))5   8.662e-01  1.177e+01   0.074  0.94135
## as.factor(cycle(ts_TREP_data))6  -1.068e+01  1.177e+01  -0.908  0.36444
## as.factor(cycle(ts_TREP_data))7  -4.728e+00  1.177e+01  -0.402  0.68801
## as.factor(cycle(ts_TREP_data))8  -1.292e+01  1.177e+01  -1.098  0.27279
## as.factor(cycle(ts_TREP_data))9  -3.771e+01  1.177e+01  -3.205  0.00142 **
## as.factor(cycle(ts_TREP_data))10 -2.514e+01  1.183e+01  -2.126  0.03391 *
## as.factor(cycle(ts_TREP_data))11 -2.612e+01  1.183e+01  -2.209  0.02756 *
## as.factor(cycle(ts_TREP_data))12 -6.483e+00  1.183e+01  -0.548  0.58377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 59.42 on 596 degrees of freedom
## Multiple R-squared:  0.8177, Adjusted R-squared:  0.814
## F-statistic: 222.8 on 12 and 596 DF,  p-value: < 2.2e-16
```

```r
# Fit seasonal means model for HEPC
lm_seasonal_HEPC <- lm(ts_HEPC_data~time(ts_HEPC_data)+as.factor(cycle(ts_HEPC_data)))
beta_int_HEPC <- lm_seasonal_HEPC$coefficients[1]
beta_coeff_HEPC <- lm_seasonal_HEPC$coefficients[2:13]
summary(lm_seasonal_HEPC)
```

```
##
## Call:
## lm(formula = ts_HEPC_data ~ time(ts_HEPC_data) + as.factor(cycle(ts_HEPC_data)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.751  -5.828  -0.399   5.746  32.175
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   316.10190   56.74394   5.571 3.84e-08 ***
## time(ts_HEPC_data)             -0.11562    0.02839  -4.072 5.28e-05 ***
## as.factor(cycle(ts_HEPC_data))2   -7.52293    2.03251  -3.701 0.000234 ***
## as.factor(cycle(ts_HEPC_data))3    2.03700    2.03251   1.002 0.316650
## as.factor(cycle(ts_HEPC_data))4    0.54065    2.03252   0.266 0.790331
## as.factor(cycle(ts_HEPC_data))5    9.15346    2.03253   4.503 8.04e-06 ***
## as.factor(cycle(ts_HEPC_data))6    5.89086    2.03254   2.898 0.003890 **
## as.factor(cycle(ts_HEPC_data))7   -0.83723    2.03255  -0.412 0.680554
## as.factor(cycle(ts_HEPC_data))8  -10.41677    2.03257  -5.125 4.03e-07 ***
## as.factor(cycle(ts_HEPC_data))9  -21.52760    2.03259 -10.591  < 2e-16 ***
## as.factor(cycle(ts_HEPC_data))10 -21.24637    2.04265 -10.401  < 2e-16 ***
## as.factor(cycle(ts_HEPC_data))11 -15.65354    2.04266  -7.663 7.38e-14 ***
## as.factor(cycle(ts_HEPC_data))12  -4.75912    2.04268  -2.330 0.020147 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.26 on 596 degrees of freedom
## Multiple R-squared:  0.4841, Adjusted R-squared:  0.4737
## F-statistic:  46.6 on 12 and 596 DF,  p-value: < 2.2e-16
```

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?

When compared to our plots in Q1 it is extremely apparent that plots have trends farmore so than what we saw in question 1.

```
cycle_values <- cycle(ts_TBEP_data)


#Fit seasonal means model for TBEP
lm_seasonal_TBEP <- lm(ts_TBEP_data~time(ts_TBEP_data)+          as.factor(cycle(ts_TBEP_data)))
      beta_int_TBEP <- lm_seasonal_TBEP$coefficients[1]
      beta_coeff_TBEP <- lm_seasonal_TBEP$coefficients[2:13]

lm_seasonal_TREP <- lm(ts_TREP_data~time(ts_TREP_data)+         as.factor(cycle(ts_TREP_data)))
      beta_int_TREP <- lm_seasonal_TREP$coefficients[1]
      beta_coeff_TREP <- lm_seasonal_TREP$coefficients[2:13]


lm_seasonal_HEPC <- lm(ts_HEPC_data~time(ts_HEPC_data)+         as.factor(cycle(ts_HEPC_data)))
      beta_int_HEPC <- lm_seasonal_HEPC$coefficients[1]
      beta_coeff_HEPC <- lm_seasonal_HEPC$coefficients[2:13]

#using modulo operator
cycle_values_TBEP <-cycle(ts_TBEP_data)
     seasonal_component_TBEP <- beta_int_TBEP+beta_coeff_TBEP*(cycle_values-1)    %% length(beta_coeff_
```

```
## Warning in `*.default`(beta_coeff_TBEP, (cycle_values -
## 1)%%length(beta_coeff_TBEP)): longer object length is not a multiple of shorter
## object length
```

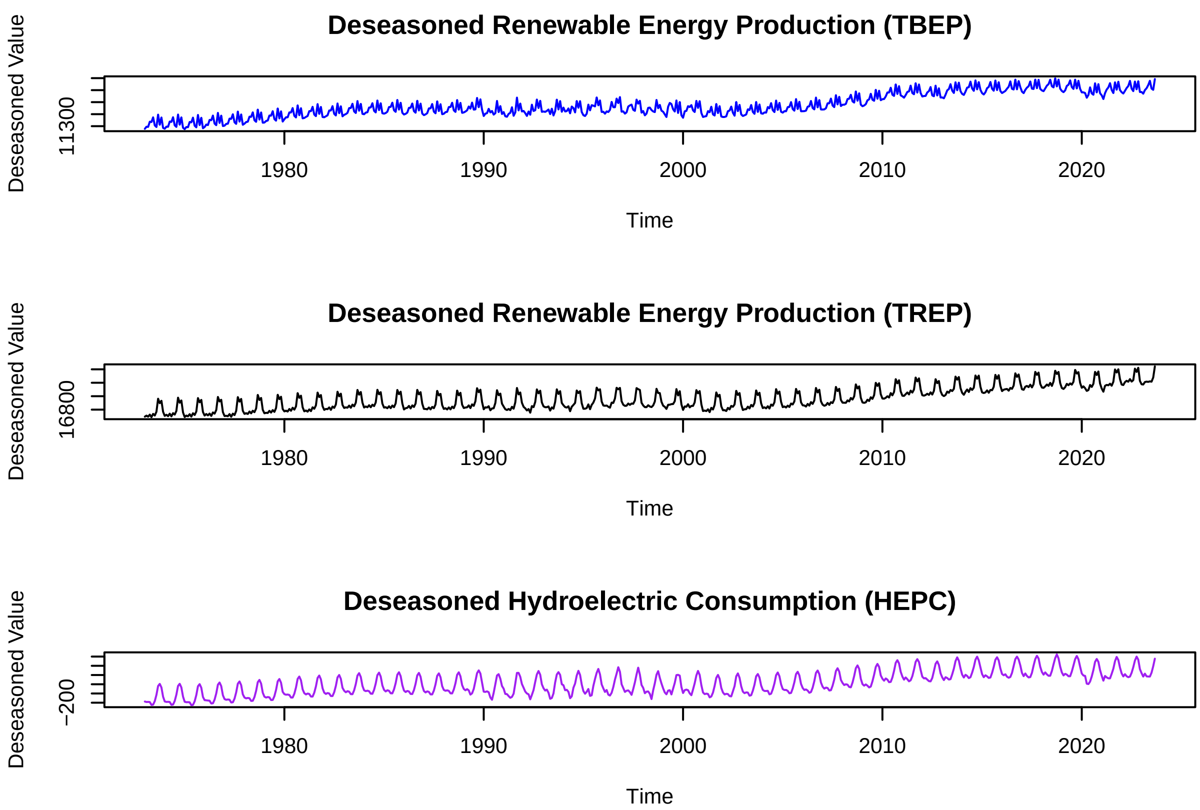
```
cycle_values_TREP <-cycle(ts_TREP_data)
      seasonal_component_TREP <- beta_int_TREP+beta_coeff_TREP*(cycle_values-1)    %% length(beta_coeff
```

```
## Warning in `*.default`(beta_coeff_TREP, (cycle_values -
## 1)%%length(beta_coeff_TREP)): longer object length is not a multiple of shorter
## object length
```

```
cycle_values_HEPC <-cycle(ts_HEPC_data)
      seasonal_component_HEPC <- beta_int_HEPC+beta_coeff_HEPC*(cycle_values-1)    %% length(beta_coeff
```

```
## Warning in `*.default`(beta_coeff_HEPC, (cycle_values -
## 1)%%length(beta_coeff_HEPC)): longer object length is not a multiple of shorter
## object length
```

```r
# Deseasoning  TBEP series
deseasoned_TBEP <-ts_TBEP_data-seasonal_component_TBEP

deseasoned_TREP <-ts_TREP_data-seasonal_component_TREP

deseasoned_HEPC <-ts_TBEP_data-seasonal_component_HEPC

# Set up a 3-row, 1-column layout for the plot
par(mfrow = c(3, 1))

#Plot of deseasoned TBEP series
plot(deseasoned_TBEP, main = "Deseasoned Renewable Energy Production (TBEP)", ylab = "Deseasoned Value"

plot(deseasoned_TREP, main = "Deseasoned Renewable Energy Production (TREP)", ylab = "Deseasoned Value"

plot(deseasoned_HEPC, main = "Deseasoned Hydroelectric Consumption (HEPC)", ylab = "Deseasoned Value", 
```



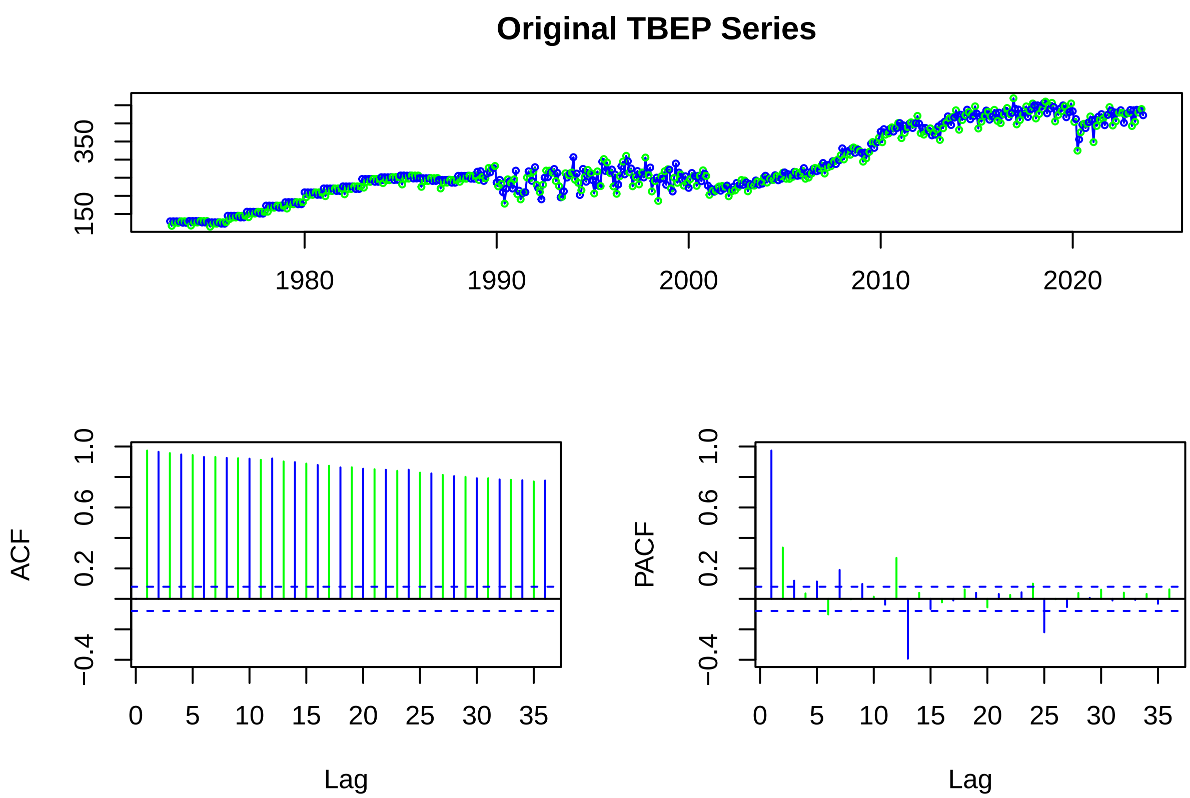

```r
# Reset the layout
par(mfrow = c(1, 1))
```

Q9

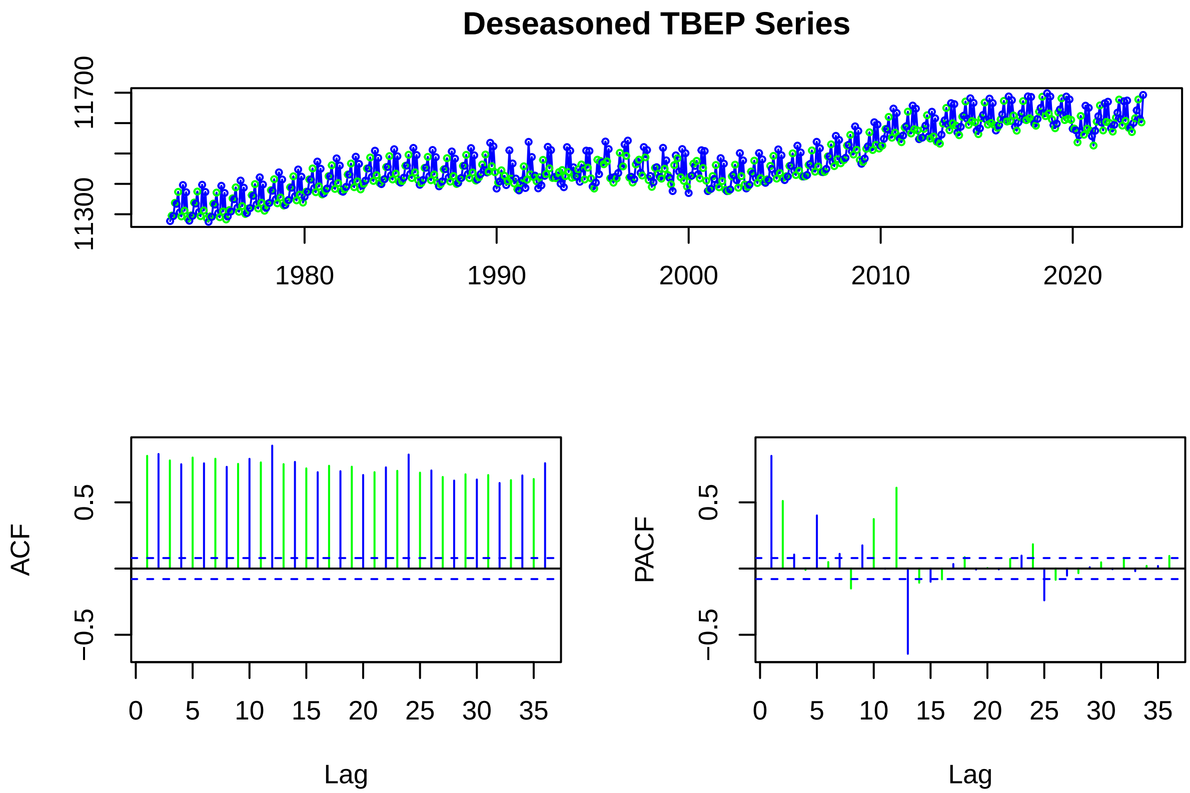
Plot ACF and PACF for the deseason series and compare with the plots from Q1. You may use plot_grid() again to get them side by side. nut mot mandatory. Did the plots change? How?

You are able to see the change for all the graphs there are a lot more cyclicality which is seen through all the graphs of TS, ACF, PACF for detrended data.

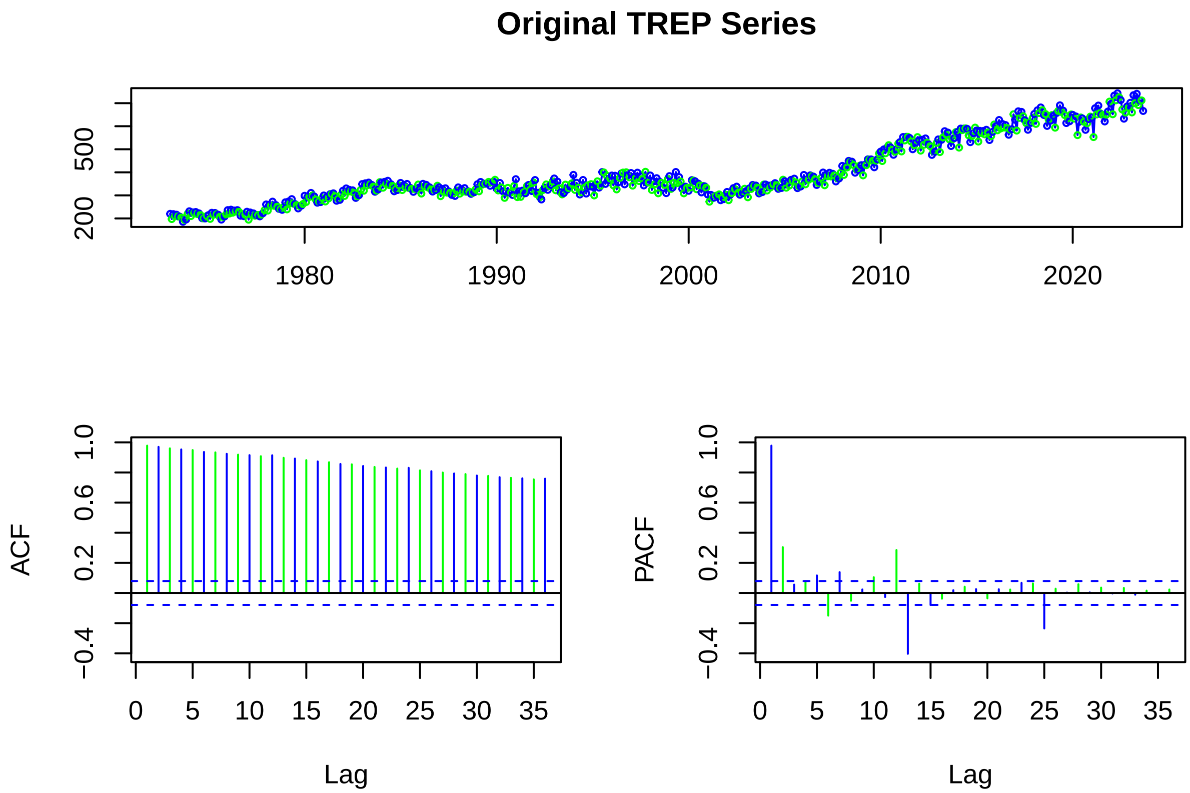
```
###This code uses the tsdisplay function to show time series components,ACF and PACF plots.

#Display time series components for original and deseasoned series
tsdisplay(ts_TBEP_data, main = "Original TBEP Series", col = c("blue", "green"))
```



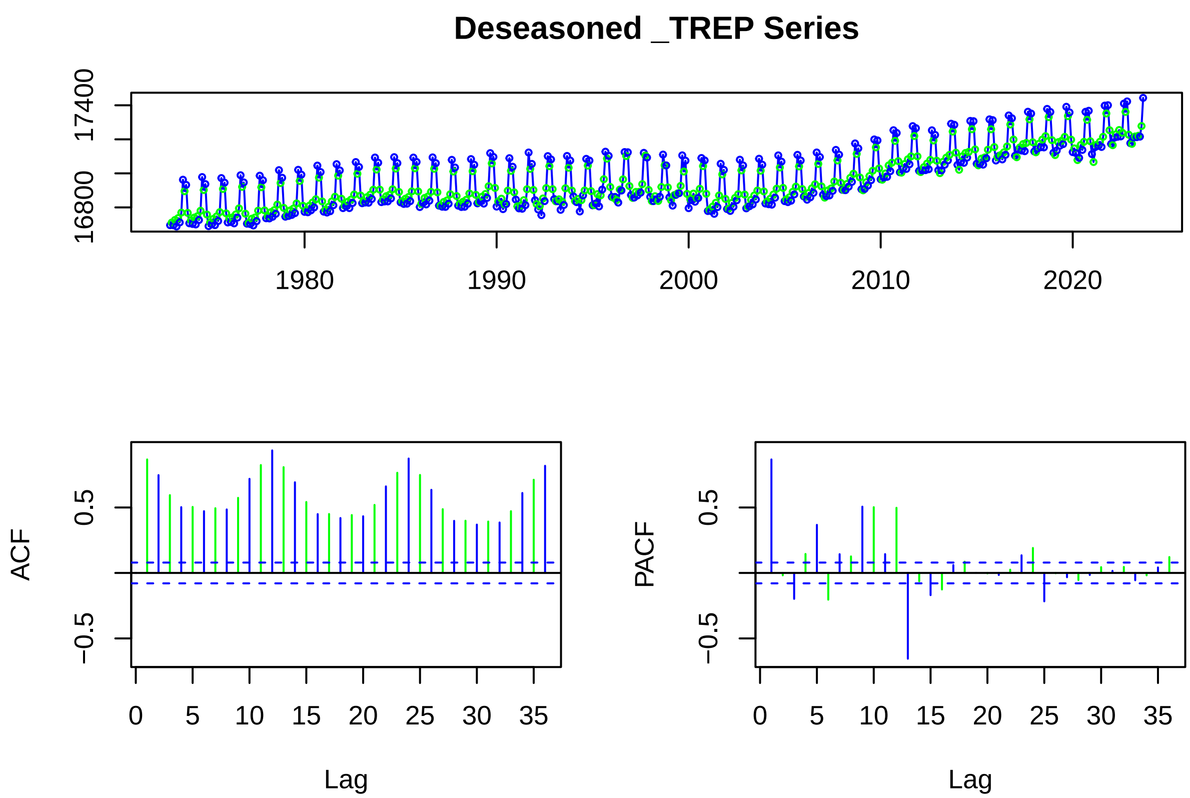

```
tsdisplay(deseasoned_TBEP, main = "Deseasoned TBEP Series", col = c("blue", "green"))
```



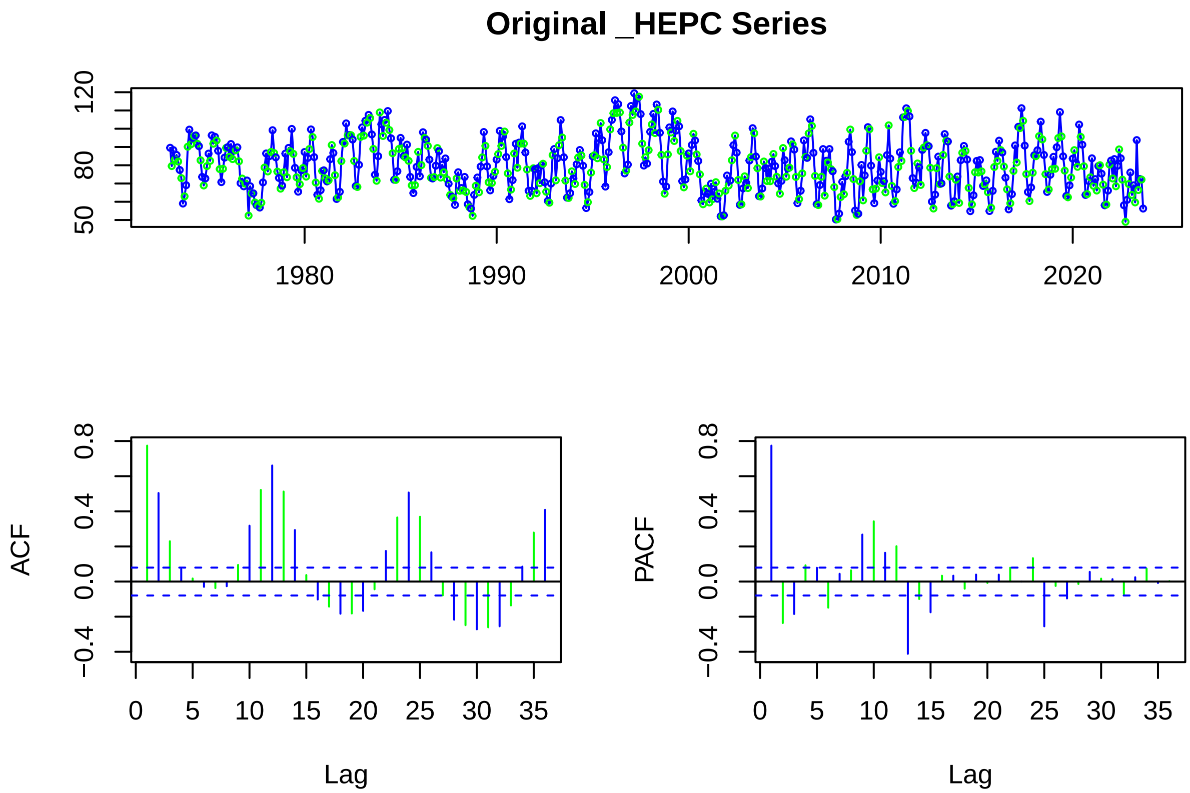

```r
#TREP data
tsdisplay(ts_TREP_data, main = "Original TREP Series", col = c("blue", "green"))
```



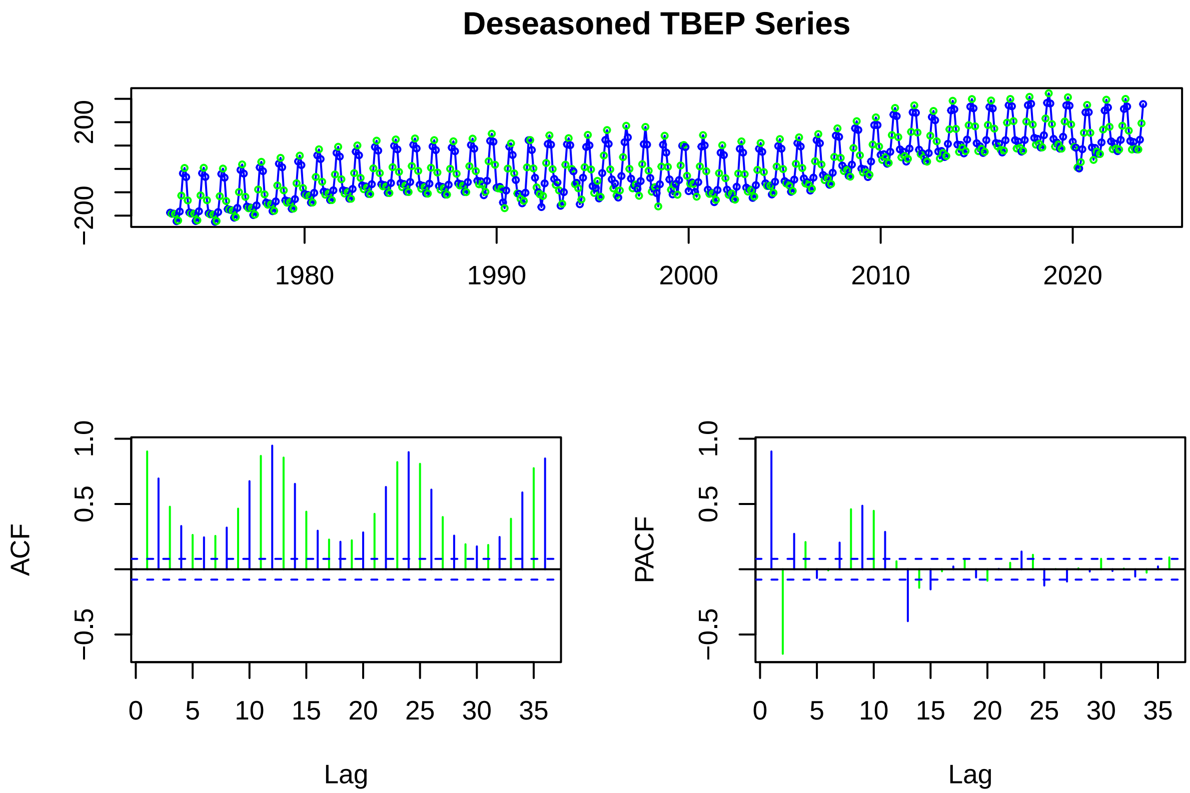

```r
tsdisplay(deseasoned_TREP, main = "Deseasoned _TREP Series", col = c("blue", "green"))
```




```r
#HEPC data
tsdisplay(ts_HEPC_data, main = "Original _HEPC Series", col = c("blue", "green"))
```




```r
tsdisplay(deseasoned_HEPC, main = "Deseasoned TBEP Series", col = c("blue", "green"))
```


Deseasoned TBEP Series
200
−200
1980
1990
2000
2010
2020
ACF
1.0
0.5
−0.5
0
5
10
15
20
25
30
35
Lag
PACF
1.0
0.5
−0.5
0
5
10
15
20
25
30
35
Lag