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

Assignment 3 - Due date 02/01/24

Ina Liao

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_Consumption
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.\

```
#install.packages("lubridate")
#install.packages("ggplot2")
#install.packages("forecast")
#install.packages("here")
#install.packages("patchwork")
#install.packages("tidyr")
#install.packages("knitr")
#install.packages("kableExtra")
#install.packages("ggthemes")
#install.packages("cowplot")
library(lubridate)
library(ggplot2)
library(forecast) #added for Acf and Pacf functions
library(here)
```

```r
library(patchwork)
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.3.2
```

```r
library(knitr)
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 4.3.2
```

```r
library(ggthemes)
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 4.3.2
```

```r
#check working directory
here()

#import data
raw_energy<-read.csv(here("Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.csv"),s
raw_energy

#transform date format
Date<-ym(raw_energy$Month)
raw_energy<-cbind(Date,raw_energy[,2:14])
raw_energy
```

```r
df_energy<-raw_energy[,c(1,5,6)]
head(df_energy)
```

```
##         Date Total.Renewable.Energy.Production Hydroelectric.Power.Consumption
## 1 1973-01-01                           219.839                          89.562
## 2 1973-02-01                           197.330                          79.544
## 3 1973-03-01                           218.686                          88.284
## 4 1973-04-01                           209.330                          83.152
## 5 1973-05-01                           215.982                          85.643
## 6 1973-06-01                           208.249                          82.060
```

```r
#rename column names
new_names<-c("Date","Renewable Production","Hydroelectric Consumption")
colnames(df_energy)<-new_names
head(df_energy)
```

```
##         Date Renewable Production Hydroelectric Consumption
## 1 1973-01-01              219.839                    89.562
## 2 1973-02-01              197.330                    79.544
## 3 1973-03-01              218.686                    88.284
## 4 1973-04-01              209.330                    83.152
## 5 1973-05-01              215.982                    85.643
## 6 1973-06-01              208.249                    82.060
```

```r
#check if there is any missing data
missing_data<-any(is.na(df_energy))
missing_data #there is no missing data in the dataframe
```

```
## [1] FALSE
```

```r
#find the start date
year1<-year(df_energy$Date[1])
month1<-month(df_energy$Date[1])

ts_energy<-ts(df_energy,start=c(year1,month1),frequency=12)
```

##Trend Component

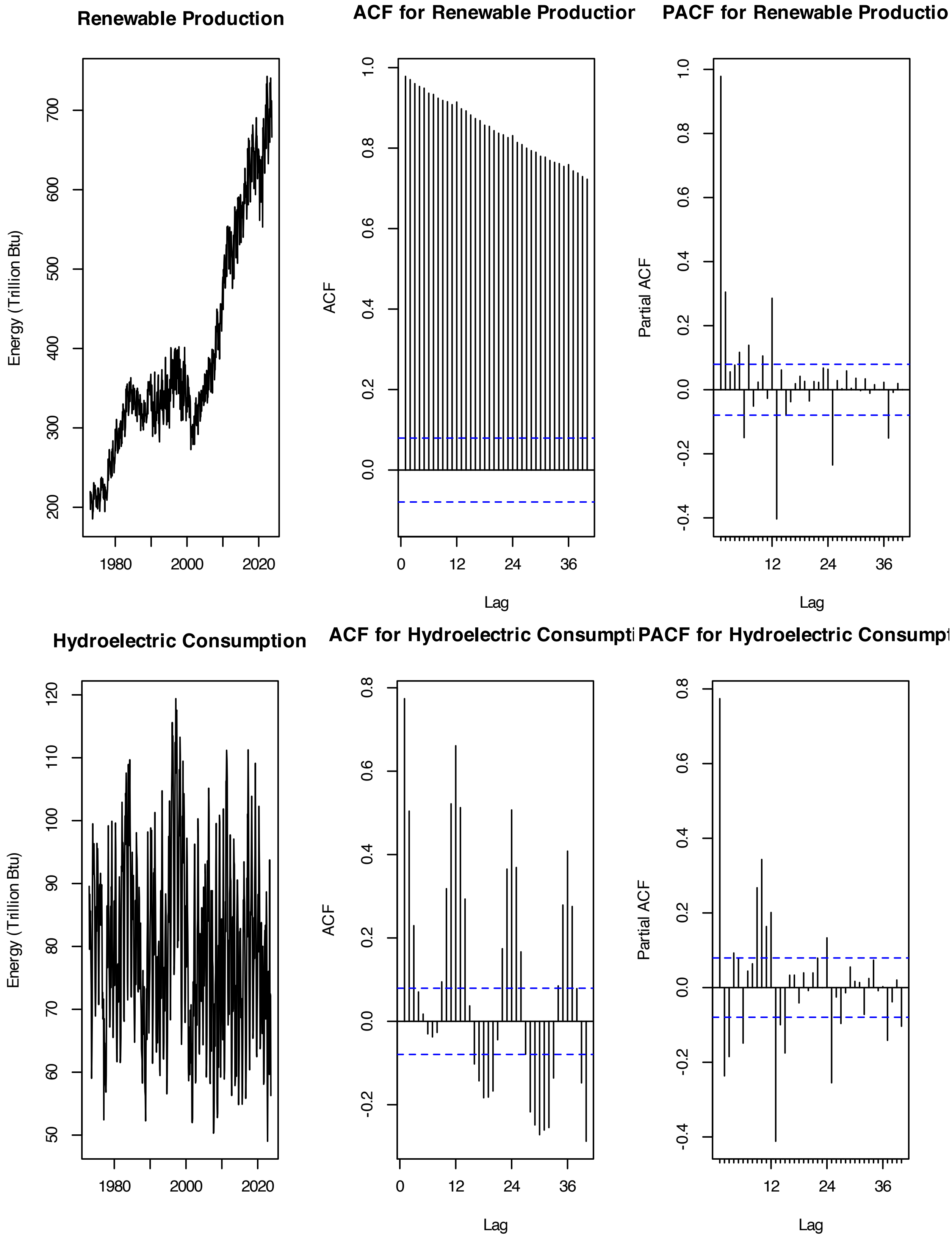
Q1

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)

```r
my_plot_theme<- theme_calc()+
  theme(
    #plot background
    plot.background = element_rect(color="gray"),

    #plot title
    plot.title=element_text(color="black",hjust=0.5,vjust=1),

    #axis labels
    axis.title.x = element_text(size = 8),
    axis.title.y = element_text(size = 8),

    #gridlines
    panel.grid.major=element_line("white"),
    panel.grid.minor = element_line("white"),
    axis.ticks=element_line(color="gray"),

    #legend
    legend.key=element_rect(color="white"),
    legend.background = element_rect(color="white"),
    legend.text = element_text(size = 8),
    legend.position="right"
      )
theme_set(my_plot_theme)
```

```r
#for the purpose of the for loop
num_row<-nrow(df_energy)
num_col<-ncol(df_energy) #the first column is Date
```

```r
#place plot side by side
par(mfrow=c(1,3))

for(i in 2:3){ #first column of df_energy is date
plot(df_energy$Date,ts_energy[,i],type="l",col="black",
ylab="Energy (Trillion Btu)",xlab=NA,main=paste(colnames(df_energy[i]),sep=""))
Acf(ts_energy[,i],lag.max=40,main=paste("ACF for",colnames(df_energy[i]),sep=" "))
Pacf(ts_energy[,i],lag.max=40,main=paste("PACF for",colnames(df_energy[i]),sep=" "))
}
```


Renewable Production
Energy (Trillion Btu)
200
300
400
500
600
700
1980
2000
2020
ACF for Renewable Productior
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0
12
24
36
Lag
PACF for Renewable Productio
Partial ACF
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
12
24
36
Lag
Hydroelectric Consumption
Energy (Trillion Btu)
50
60
70
80
90
100
110
120
1980
2000
2020
ACF for Hydroelectric Consumpti
ACF
-0.2
0.0
0.2
0.4
0.6
0.8
0
12
24
36
Lag
PACF for Hydroelectric Consumpt
Partial ACF
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
12
24
36
Lag

```r
#use "plot" in the for loop instead of print(ggplot)
```

Q2

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

Renewable energy production has been increasing over time, but there is no clear seasonal variation in the trend. Hydroelectric power consumption does not show any noticeable growth or decline over time, but it does have seasonal components in the data.

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
#create a time vector
t<-c(1:num_row)

#create a new data frame
df_renew<-data.frame("time"=t,"renewable"=df_energy$`Renewable Production`)

#run linear regression
linear_renew=lm(renewable~t,df_renew)
summary(linear_renew)
```

```
##
## Call:
## lm(formula = renewable ~ t, data = df_renew)
##
## 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) 180.98940    4.90151   36.92   <2e-16 ***
## t             0.70404    0.01392   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
#store the intercept and slope
intercept_renew=as.numeric(linear_renew$coefficients[1])
slope_renew=as.numeric(linear_renew$coefficients[2])
```

```r
#create a new data frame
df_hydro<-data.frame("time"=t,"hydro"=df_energy$`Hydroelectric Consumption`)

#run linear regression
linear_hydro=lm(hydro~t,df_hydro)
summary(linear_hydro)
```

```
##
## Call:
## lm(formula = hydro ~ t, data = df_hydro)
##
## 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) 82.734747   1.140265  72.557  < 2e-16 ***
## t           -0.009849   0.003239  -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
```

```
#store the intercept and slope
intercept_hydro=as.numeric(linear_hydro$coefficients[1])
slope_hydro=as.numeric(linear_hydro$coefficients[2])
```

Q4

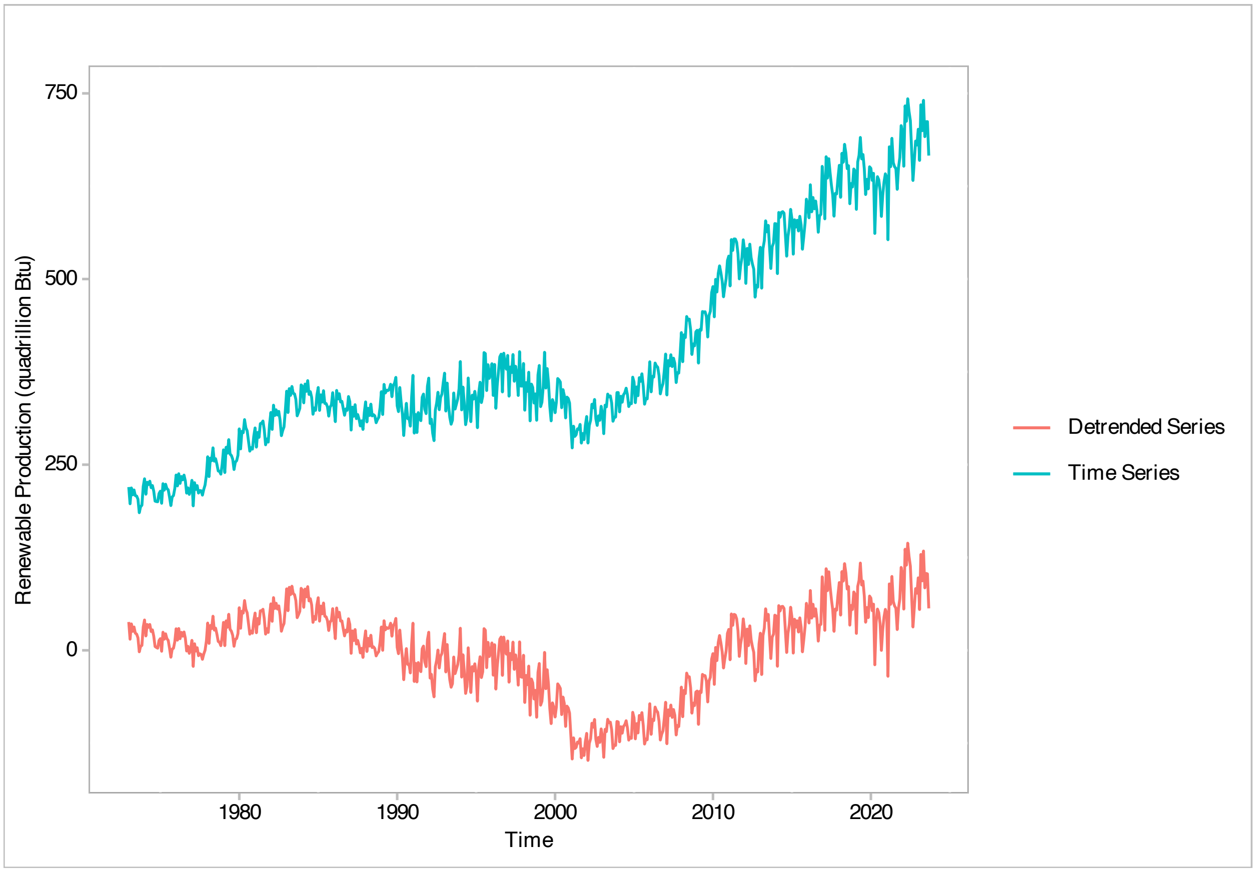
Use the regression coefficients from Q3 to detrend the series. Plot the detrended series and compare with the plots from Q1. What happened? Did anything change?

```
#create the detrended time series from linear trend
detrend_renew<-df_energy$`Renewable Production`-
  (slope_renew*df_renew$time+intercept_renew)

#create detrended time series object
ts_detrend_renew<-ts(detrend_renew,start=c(year1,month1),frequency=12)
```

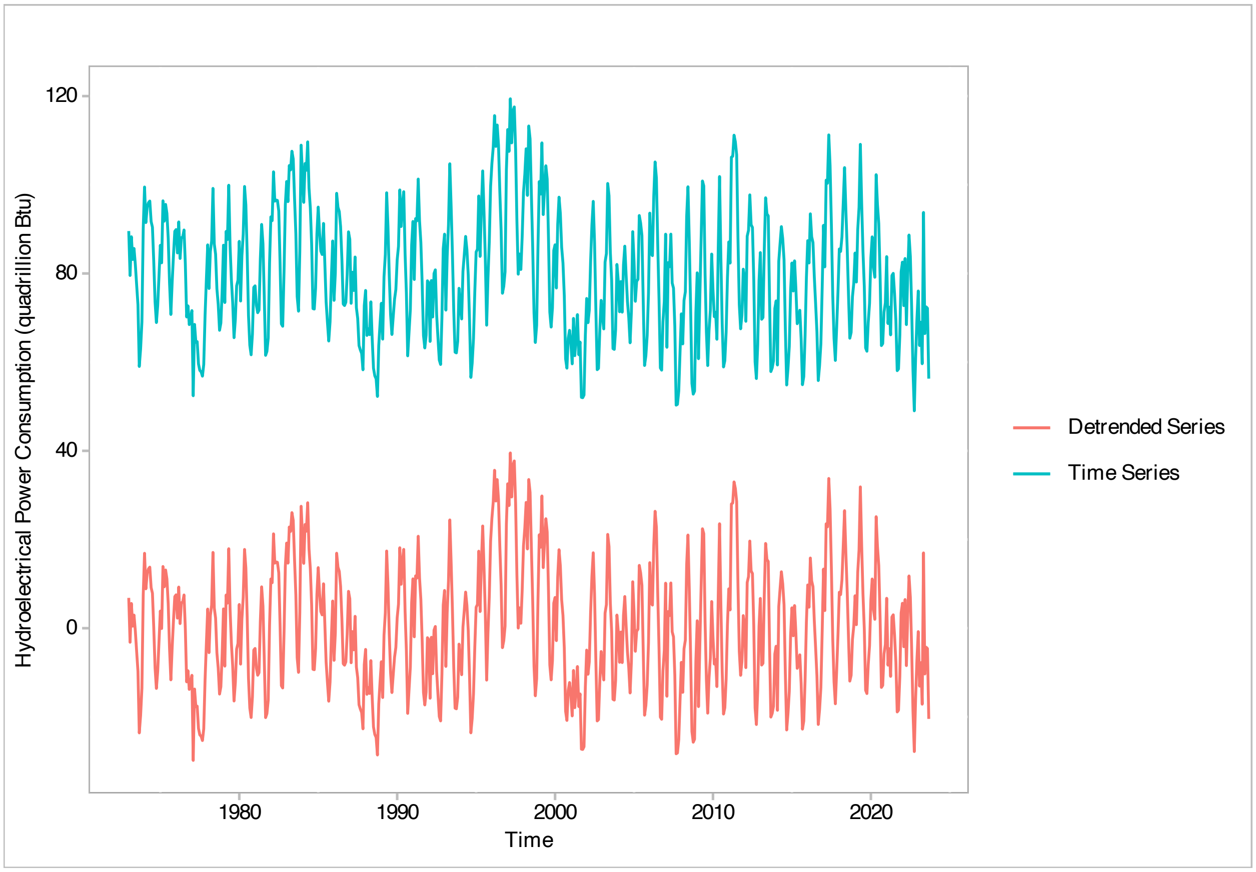
```
#create the detrended time series from linear trend
detrend_hydro<-df_energy$`Hydroelectric Consumption`-
  (slope_hydro*df_hydro$time+intercept_hydro)

#create detrended time series object
ts_detrend_hydro<-ts(detrend_hydro,start=c(year1,month1),frequency=12)
```

```
#combine detrended time series object
ts_detrend_all<-cbind(t,ts_detrend_renew,ts_detrend_hydro)
```

```
plot_detrend_renew<-autoplot(ts_energy[,2],series="Time Series")+
  autolayer(ts_detrend_all[,2], series="Detrended Series")+
  labs(x="Time",y="Renewable Production (quadrillion Btu)",color="")
plot_detrend_renew
```




```
plot_detrend_hydro<-autoplot(ts_energy[,3],series="Time Series")+
  autolayer(ts_detrend_all[,3], series="Detrended Series")+
  labs(x="Time",y="Hydroelectrical Power Consumption (quadrillion Btu)",color="")
plot_detrend_hydro
```



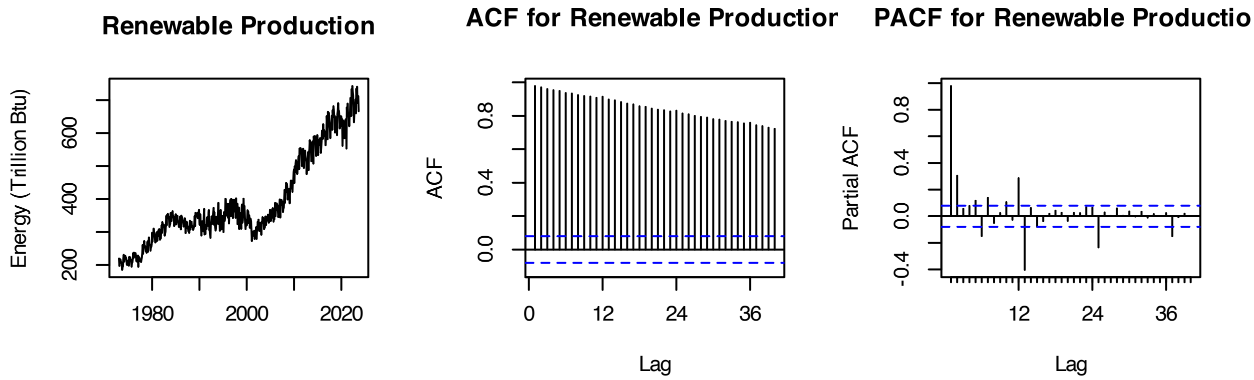
After removing linear trends, detrended renewable energy production show less upward trend over time, which might because of the underlying trend in renewable energy production being approximately linear. Both original and detrended hydroelectric power consumption do not show any significant increase or decrease over the time period.

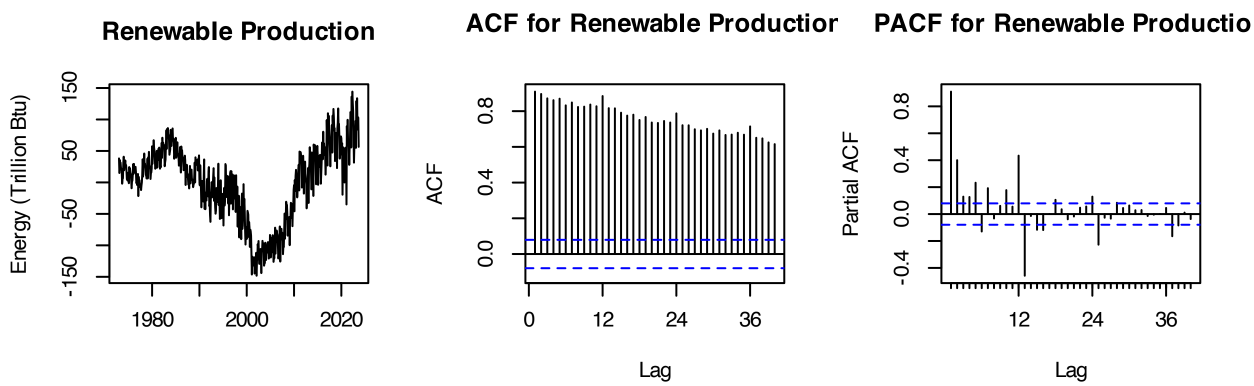
Q5

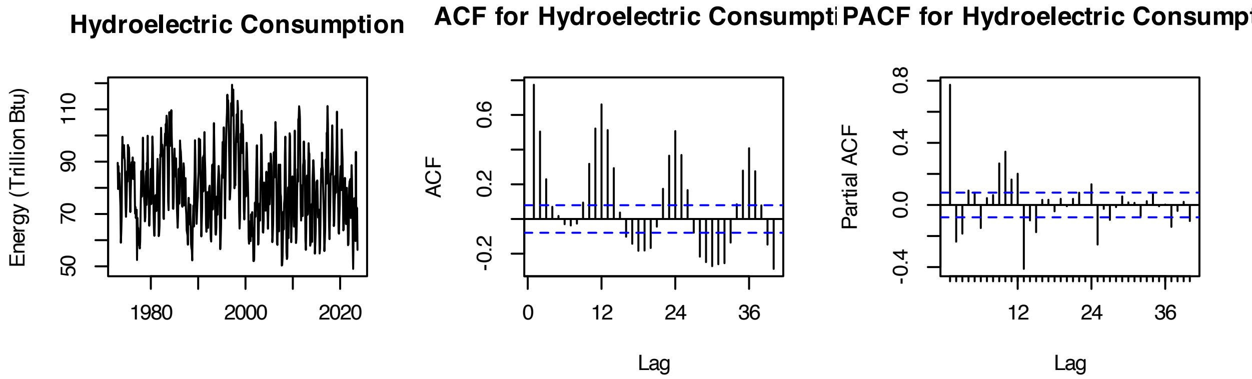
Plot ACF and PACF for the detrended series and compare with the plots from Q1. You may use plot_grid() again to get them side by side. nut mot mandatory. Did the plots change? How?

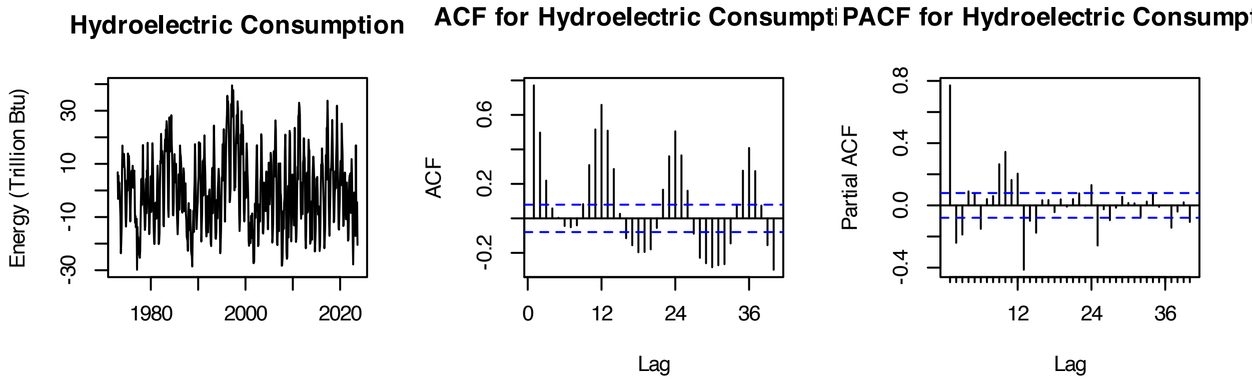
```
#place the plot side by side
par(mfrow=c(2,3))

for(i in 2:3){
  #original time series
  plot(df_energy$Date,ts_energy[,i],type="l",col="black",
       ylab="Energy (Trillion Btu)",xlab=NA,
       main=paste(colnames(df_energy[i]),sep=""))
  Acf(ts_energy[,i],lag.max=40,
      main=paste("ACF for",colnames(df_energy[i]),sep=" "))
  Pacf(ts_energy[,i],lag.max=40,
       main=paste("PACF for",colnames(df_energy[i]),sep=" "))

  #detrended time series
  plot(df_energy$Date,ts_detrend_all[,i],type="l",col="black",
       ylab="Energy (Trillion Btu)",xlab=NA,
       main=paste(colnames(df_energy[i]),sep=""))
```

```r
  Acf(ts_detrend_all[,i],lag.max=40,
      main=paste("ACF for",colnames(df_energy[i]),sep=" "))
  Pacf(ts_detrend_all[,i],lag.max=40,
       main=paste("PACF for",colnames(df_energy[i]),sep=" "))
}
```









The autocorrelation was not significantly affected by removing trends in renewable energy production and hydroelectric power consumption. This may becasue of (1) the presence of nonlinear trends in the data, which cannot be removed by a linear detrending process; and (2) the presence of seasonal components in the data.

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.

From the time series and ACF plot, we can not see a significant seasonal trend in the renewable energy production. However, hydroelectric power consumption shows a significant seasonal trend.

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?

```r
#create the seasonal dummies
seasonal_dummies_renew<-seasonaldummy(ts_detrend_renew)

#fit the detrended series to the seasonal dummies
seasonal_mean_renew=lm(detrend_renew~seasonal_dummies_renew)
summary(seasonal_mean_renew)
```

```
##
```

```
## Call:
## lm(formula = detrend_renew ~ seasonal_dummies_renew)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -146.18  -37.88   14.16   42.18  128.82
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   8.101      8.397   0.965  0.33506
## seasonal_dummies_renewJan     6.482     11.816   0.549  0.58350
## seasonal_dummies_renewFeb   -31.660     11.816  -2.679  0.00758 **
## seasonal_dummies_renewMar     6.041     11.816   0.511  0.60937
## seasonal_dummies_renewApr    -7.287     11.816  -0.617  0.53766
## seasonal_dummies_renewMay     7.349     11.816   0.622  0.53422
## seasonal_dummies_renewJun    -4.198     11.816  -0.355  0.72253
## seasonal_dummies_renewJul     1.755     11.816   0.149  0.88195
## seasonal_dummies_renewAug    -6.434     11.816  -0.544  0.58630
## seasonal_dummies_renewSep   -31.231     11.816  -2.643  0.00843 **
## seasonal_dummies_renewOct   -18.660     11.875  -1.571  0.11662
## seasonal_dummies_renewNov   -19.642     11.875  -1.654  0.09864 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 59.37 on 597 degrees of freedom
## Multiple R-squared:  0.04977,    Adjusted R-squared:  0.03226
## F-statistic: 2.843 on 11 and 597 DF,  p-value: 0.001226
```

```r
#create the seasonal dummies
seasonal_dummies_hydro<-seasonaldummy(ts_detrend_hydro)

#fit the detrended series to the seasonal dummies
seasonal_mean_hydro=lm(detrend_hydro~seasonal_dummies_hydro)
summary(seasonal_mean_hydro)
```

```
##
## Call:
## lm(formula = detrend_hydro ~ seasonal_dummies_hydro)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.805  -5.817  -0.408   5.770  32.172
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.5609     1.4503   0.387 0.699080
## seasonal_dummies_hydroJan    4.7581     2.0409   2.331 0.020070 *
## seasonal_dummies_hydroFeb   -2.7647     2.0409  -1.355 0.176057
## seasonal_dummies_hydroMar    6.7955     2.0409   3.330 0.000923 ***
## seasonal_dummies_hydroApr    5.2993     2.0409   2.597 0.009649 **
## seasonal_dummies_hydroMay   13.9124     2.0409   6.817 2.29e-11 ***
## seasonal_dummies_hydroJun   10.6500     2.0409   5.218 2.50e-07 ***
## seasonal_dummies_hydroJul    3.9221     2.0409   1.922 0.055118 .
## seasonal_dummies_hydroAug   -5.6572     2.0409  -2.772 0.005747 **
## seasonal_dummies_hydroSep  -16.7678     2.0409  -8.216 1.31e-15 ***
```

```
## seasonal_dummies_hydroOct -16.4877     2.0510  -8.039 4.88e-15 ***
## seasonal_dummies_hydroNov -10.8946     2.0510  -5.312 1.53e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.26 on 597 degrees of freedom
## Multiple R-squared:  0.4762, Adjusted R-squared:  0.4665
## F-statistic: 49.34 on 11 and 597 DF,  p-value: < 2.2e-16
```

Hydroelectric power consumption show a clear seasonal trend. The null hypothesis assumes the absence of seasonality, while the alternative hypothesis proposes its presence. Based on the results of F-test, we have have sufficient evidence to reject the null hypothesis for hydroelectric power consumption (p-value<0.001), indicating the existence of seasonality. However, we do not have sufficient evidence to reject the null hypothesis for renewable energy production (p-value>0.001).

Q8

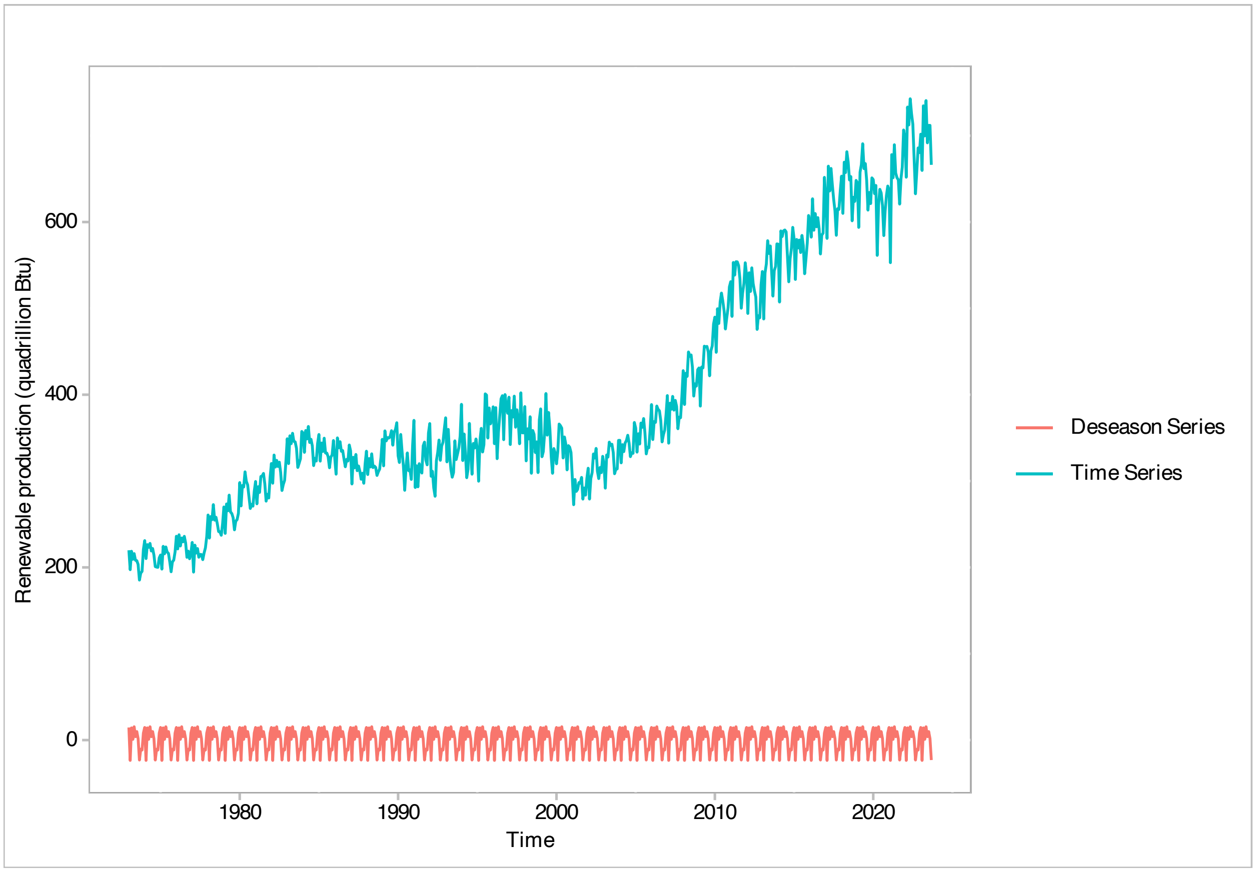
Use the regression coefficients from Q7 to deseason the series. Plot the deseason series and compare with the plots from part Q1. Did anything change?

```r
#store regression coefficients
slope_renew_deseason=seasonal_mean_renew$coefficients[2:12]
intercept_renew_deseason=seasonal_mean_renew$coefficients[1]

#compute seasonal components
seasonal_components_renew=array(0,num_row)
for (i in 1:num_row){
  seasonal_components_renew[i]=
    (intercept_renew_deseason+slope_renew_deseason%*%seasonal_dummies_renew[i,])
}

#transform seasonal components into a time series object
ts_deseason_renew<-ts(seasonal_components_renew,start=c(year1,month1),frequency=12)
```

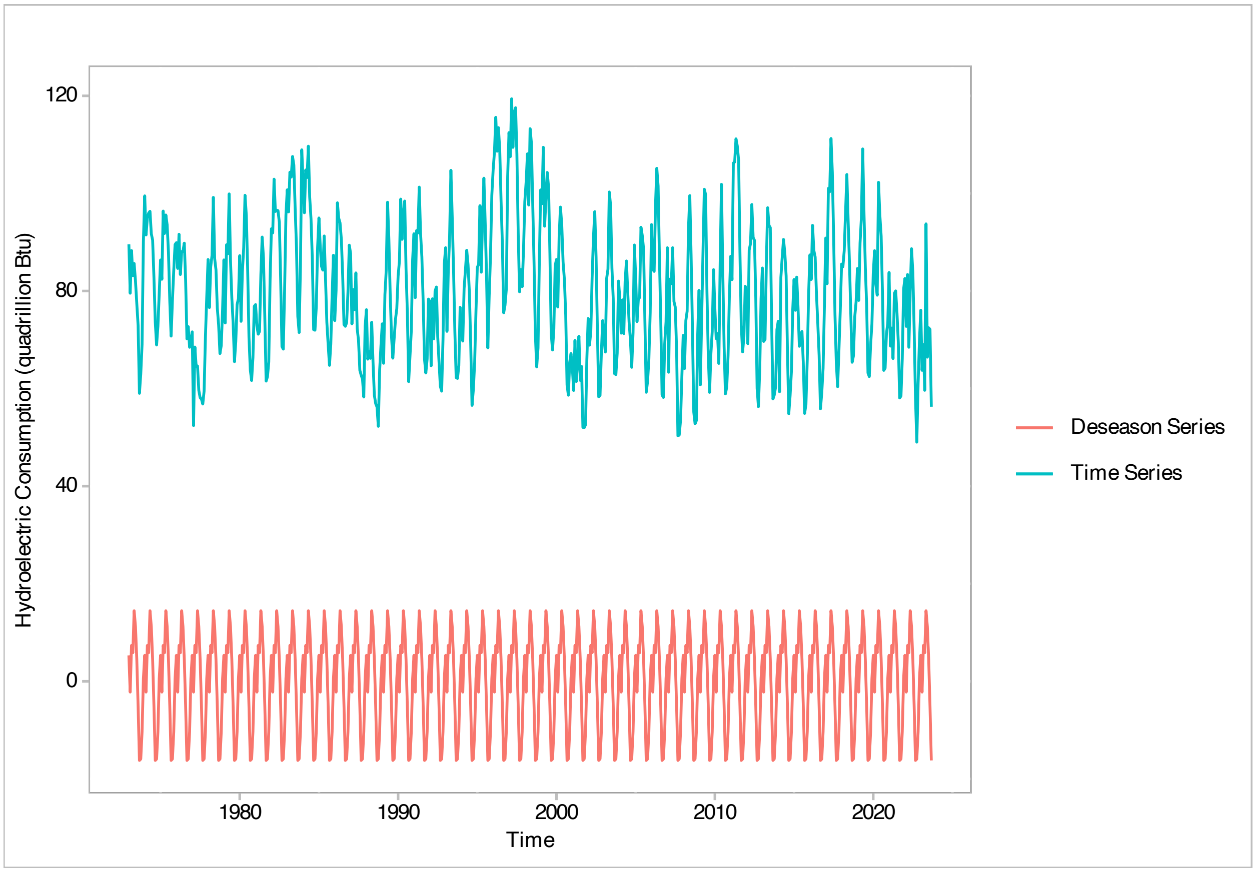
```r
#store regression coefficients
slope_hydro_deseason=seasonal_mean_hydro$coefficients[2:12]
intercept_hydro_deseason=seasonal_mean_hydro$coefficients[1]

#compute seasonal components
seasonal_components_hydro=array(0,num_row)
for (i in 1:num_row){
  seasonal_components_hydro[i]=
    (intercept_hydro_deseason+slope_hydro_deseason%*%seasonal_dummies_renew[i,])
}

#transform seasonal components into a time series object
ts_deseason_hydro<-ts(seasonal_components_hydro,start=c(year1,month1),frequency=12)
```

```r
ts_deseason_all<-cbind(t,ts_deseason_renew,ts_deseason_hydro)
```

```r
plot_deseason_renew<-autoplot(ts_energy[,2],series="Time Series")+
  autolayer(ts_deseason_all[,2], series="Deseason Series")+
  labs(x="Time",y="Renewable production (quadrillion Btu)",color="")
plot_deseason_renew
```




```r
plot_deseason_hydro<-autoplot(ts_energy[,3],series="Time Series")+
  autolayer(ts_deseason_all[,3], series="Deseason Series")+
  labs(x="Time",y="Hydroelectric Consumption (quadrillion Btu)",color="")
plot_deseason_hydro
```



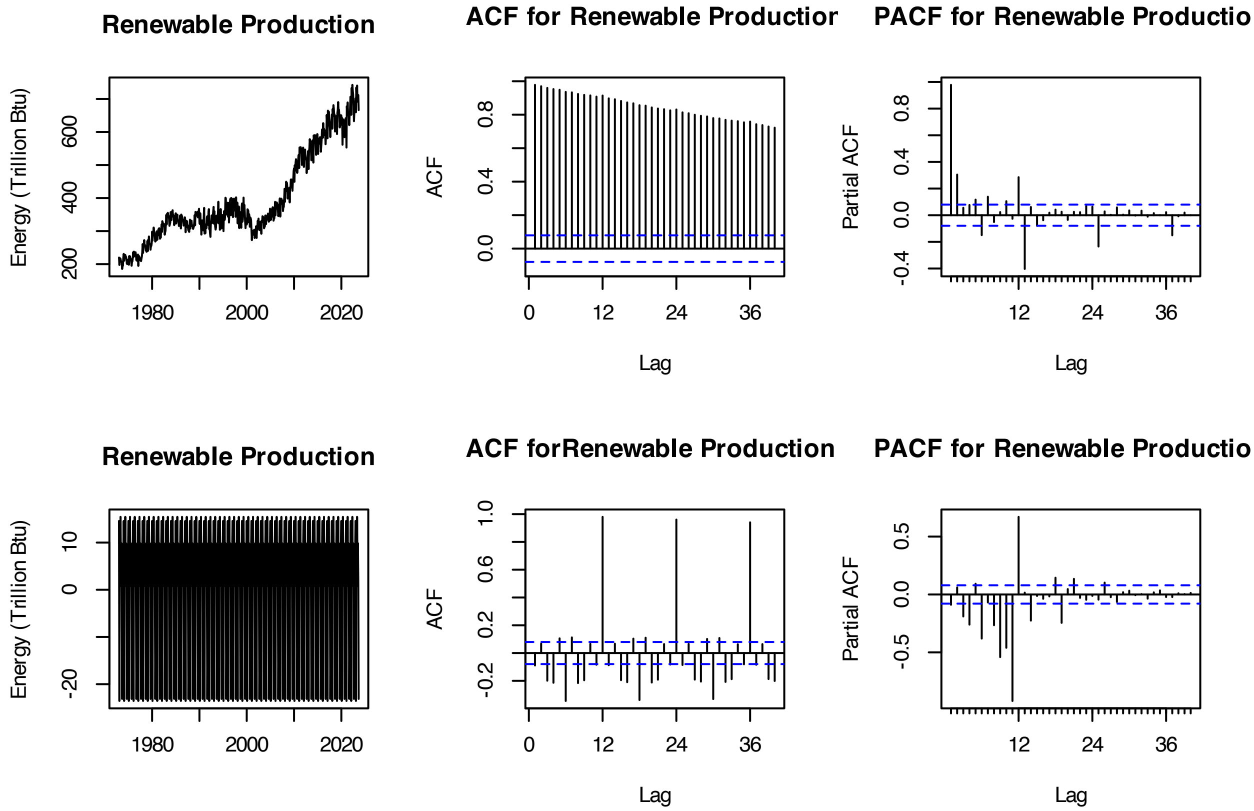
After removing seasonal components, both renewable energy production and hydroelectric power consumption show wave-like patterns with equally spaced peaks and troughs.

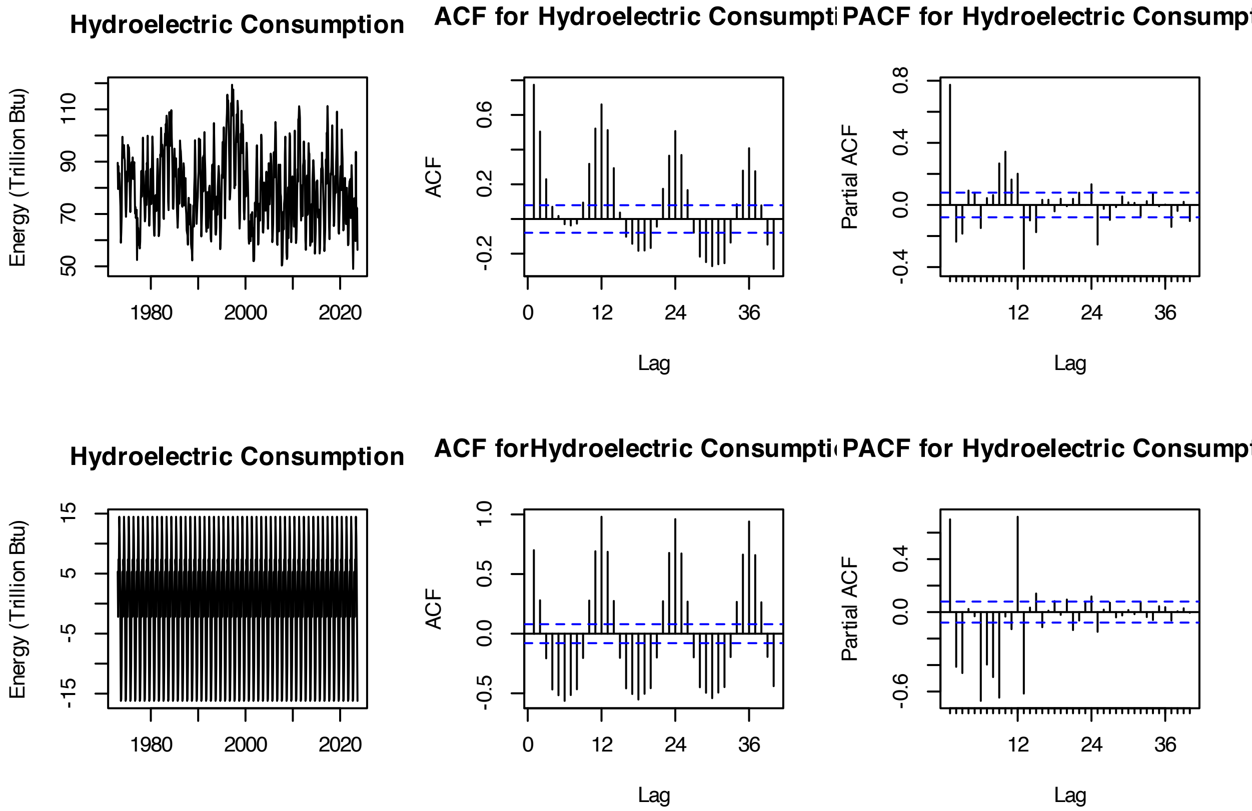
Q9

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?

```
#place the plot side by side
par(mfrow=c(2,3))

for(i in 2:3){
  #original time series
  plot(df_energy$Date,ts_energy[,i],type="l",col="black",
       ylab="Energy (Trillion Btu)",xlab=NA,
       main=paste(colnames(df_energy[i]),sep=""))
  Acf(ts_energy[,i],lag.max=40,
      main=paste("ACF for",colnames(df_energy[i]),sep=" "))
  Pacf(ts_energy[,i],lag.max=40,
       main=paste("PACF for",colnames(df_energy[i]),sep=" "))

  #detrended time series
  plot(df_energy$Date,ts_deseason_all[,i],type="l",col="black",
       ylab="Energy (Trillion Btu)",xlab=NA,
       main=paste(colnames(df_energy[i]),sep=""))
  Acf(ts_deseason_all[,i],lag.max=40,
      main=paste0("ACF for",colnames(df_energy[i]),sep=" "))
```

```
  Pacf(ts_deseason_all[,i],lag.max=40,
       main=paste("PACF for",colnames(df_energy[i]),sep=" "))
}
```





After removing the seasonal components from the renewable energy production time series data, we observed spikes at lags 13, 24, and 36 in the ACF plot. These spikes suggest that there is no clear seasonality in the data. In the hydroelectric power consumption data, we can see a clearer seasonality from the ACF plot after removing seasonal components in the data. Both renewable energy production and hydroelectric consumption have a clear cutoff at spike 13 in the PACF plot, we can thus further build a time series model that includes lags 1 and 13 to better understand the variability of the data.