ENV 790.30 - Time Series Analysis for Energy Data | Spring 2023

Assignment 3 - Due date 02/10/23

Kexin(Kassie) Huang

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_Sp23.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.\

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

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

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

##Trend Component

Q1

Create a plot window that has one row and three columns. And then for each object on your data frame, fill the plot window with time series plot, ACF and PACF. You may use the some code form A2, but I want all three plots on the same window this time. (Hint: use par() function)

Answer:

```
#import data
library(readxl)
energy_data <- read_excel(path='./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source
```

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

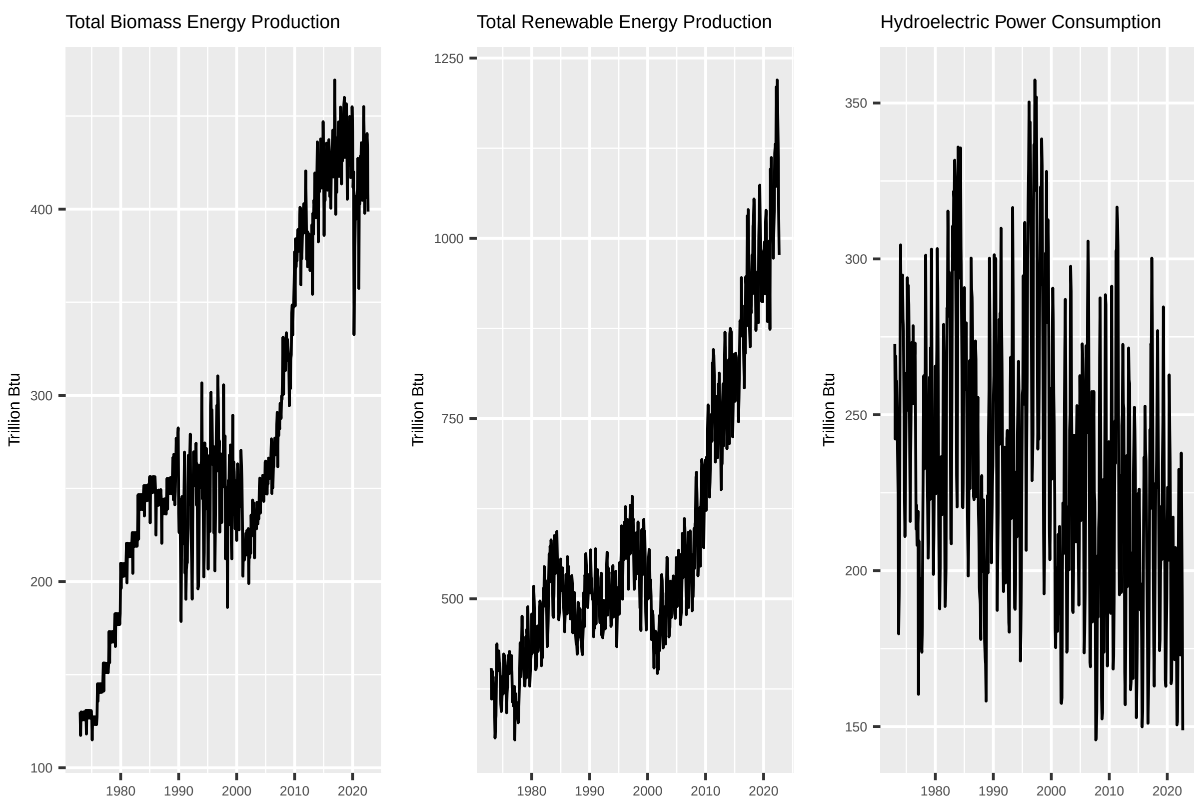
```
colnames(energy_data) <- read_excel(path='./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_
```

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

```
energy_data <- as.data.frame(energy_data[,4:6])
ts_energy_data <- ts(energy_data,start = c(1973,1), frequency = 12)

#index
datanames <- colnames(energy_data)
ncols <- ncol(energy_data)

#Time series plot
library(ggplot2)
library(ggfortify)
```

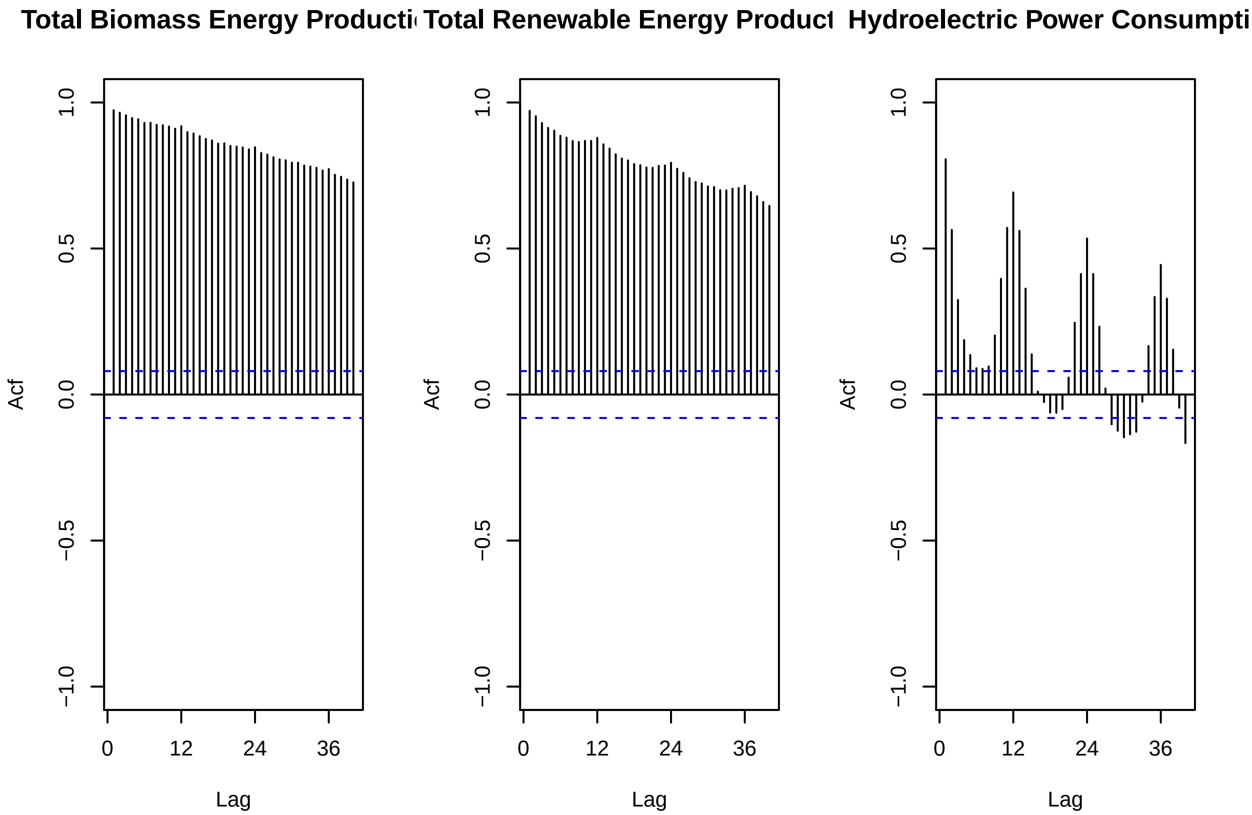
```
## Registered S3 methods overwritten by 'ggfortify':
##   method                 from
##   autoplot.Arima         forecast
##   autoplot.acf           forecast
##   autoplot.ar            forecast
##   autoplot.bats          forecast
##   autoplot.decomposed.ts forecast
##   autoplot.ets           forecast
##   autoplot.forecast      forecast
##   autoplot.stl           forecast
##   autoplot.ts            forecast
##   fitted.ar              forecast
##   fortify.ts             forecast
##   residuals.ar           forecast
```

```
library(patchwork)

plots_ts <- list()
for(i in 1:ncols){
  p <-  autoplot(ts_energy_data[,i])+
    labs(title = datanames[i],y='Trillion Btu')+
    theme(text = element_text(size = 6))
  plots_ts[[i]] <- p
}
wrap_plots(plots_ts, ncol = 3)
```

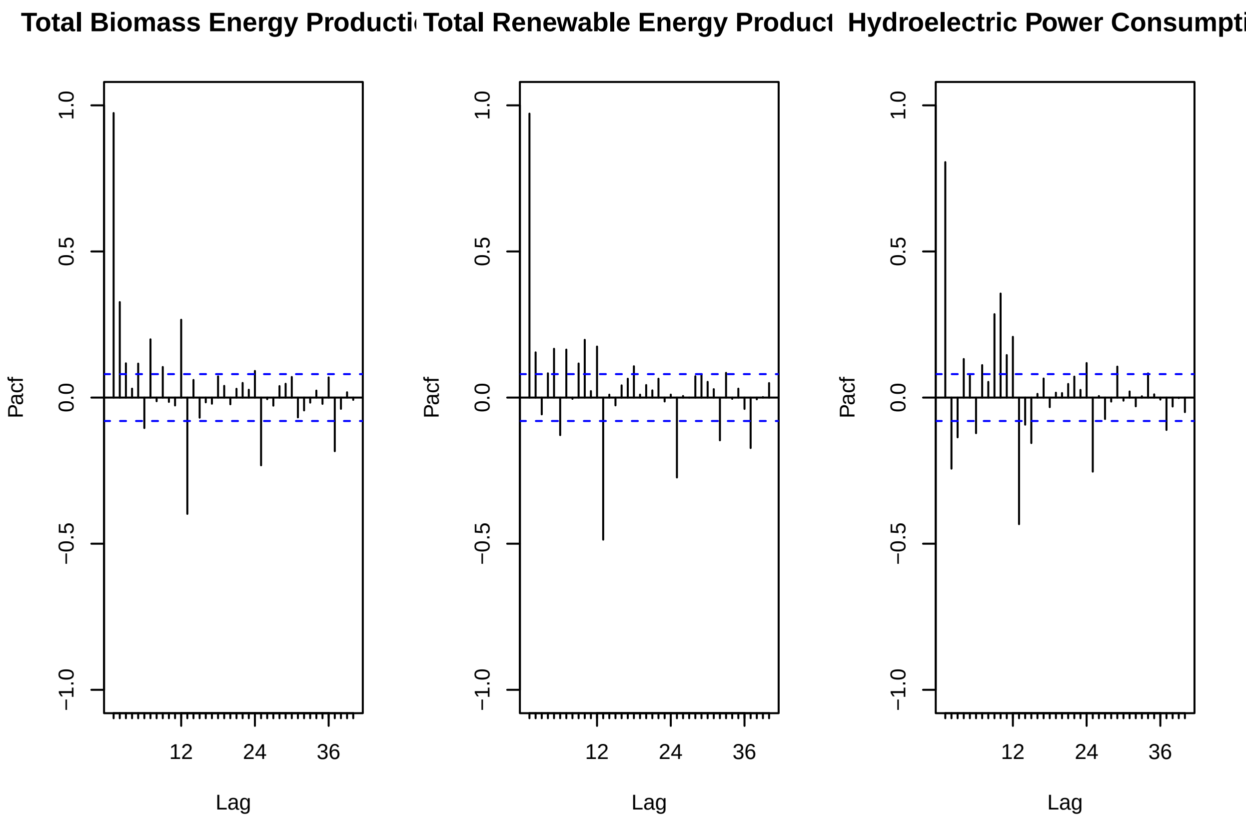



```
#ACF&PACF
par(mfrow=c(1,3))
for(i in 1:ncols){
```

```
  Acf(ts_energy_data[,i],lag.max=40,main=datanames[i],cex.main=0.5,ylim=c(-1,1),ylab='Acf')
}
```




```
for(i in 1:ncols){
  Pacf(ts_energy_data[,i],lag.max=40,main=datanames[i],cex.main=0.5,ylim=c(-1,1),ylab='Pacf')
}
```



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?

Answer:

Yes, all of them have a trend. Both the total biomass energy production and the total renewable energy production have an increasing trend, and the hydroelectric power consumption has a slightly decreasing trend.

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.

Answer:

```
nobs <- nrow(energy_data)
t <- 1:nobs


#Biomass Energy Production
linear_trend_biomass <- lm(energy_data$`Total Biomass Energy Production`~t)
summary(linear_trend_biomass)
```

```
##
## Call:
## lm(formula = energy_data$`Total Biomass Energy Production` ~
##     t)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -102.800  -23.994    5.667   32.265   82.192
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.337e+02  3.245e+00   41.22   <2e-16 ***
## t           4.800e-01  9.402e-03   51.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.59 on 595 degrees of freedom
## Multiple R-squared:  0.8142, Adjusted R-squared:  0.8138
## F-statistic:  2607 on 1 and 595 DF,  p-value: < 2.2e-16
```

```
beta0_biomass <- linear_trend_biomass$coefficients[1]
beta1_biomass <- linear_trend_biomass$coefficients[2]

#Renewable Energy Production
linear_trend_renewable <- lm(energy_data$`Total Renewable Energy Production`~t)
summary(linear_trend_renewable)
```

```
##
## Call:
## lm(formula = energy_data$`Total Renewable Energy Production` ~
##     t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -238.75  -61.85    8.59   64.48  352.27
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 312.2475     8.4902   36.78   <2e-16 ***
## t             0.9362     0.0246   38.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 103.6 on 595 degrees of freedom
## Multiple R-squared:  0.7088, Adjusted R-squared:  0.7083
## F-statistic:  1448 on 1 and 595 DF,  p-value: < 2.2e-16
```

```
beta0_renewable <- linear_trend_renewable$coefficients[1]
beta1_renewable <- linear_trend_renewable$coefficients[2]

#Hydroelectric Power Consumption
linear_trend_hydro <- lm(energy_data$`Hydroelectric Power Consumption`~t)
summary(linear_trend_hydro)
```

```
##
## Call:
## lm(formula = energy_data$`Hydroelectric Power Consumption` ~
##     t)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -95.42 -31.20  -2.56  27.32 121.61
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 259.898013   3.427300  75.832  < 2e-16 ***
## t            -0.082888   0.009931  -8.346 4.94e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41.82 on 595 degrees of freedom
## Multiple R-squared:  0.1048, Adjusted R-squared:  0.1033
## F-statistic: 69.66 on 1 and 595 DF,  p-value: 4.937e-16
```

```r
beta0_hydro <- linear_trend_hydro$coefficients[1]
beta1_hydro <- linear_trend_hydro$coefficients[2]
```

Interpretation: For biomass energy production, the coefficient of t is 0.48 with the p-value less than 0.001, and the intercept is 133.7. In other words, the biomass energy production is positively correlated with time, and the linear trend model can be written as: Biomass energy production = 134 + 0.48*t

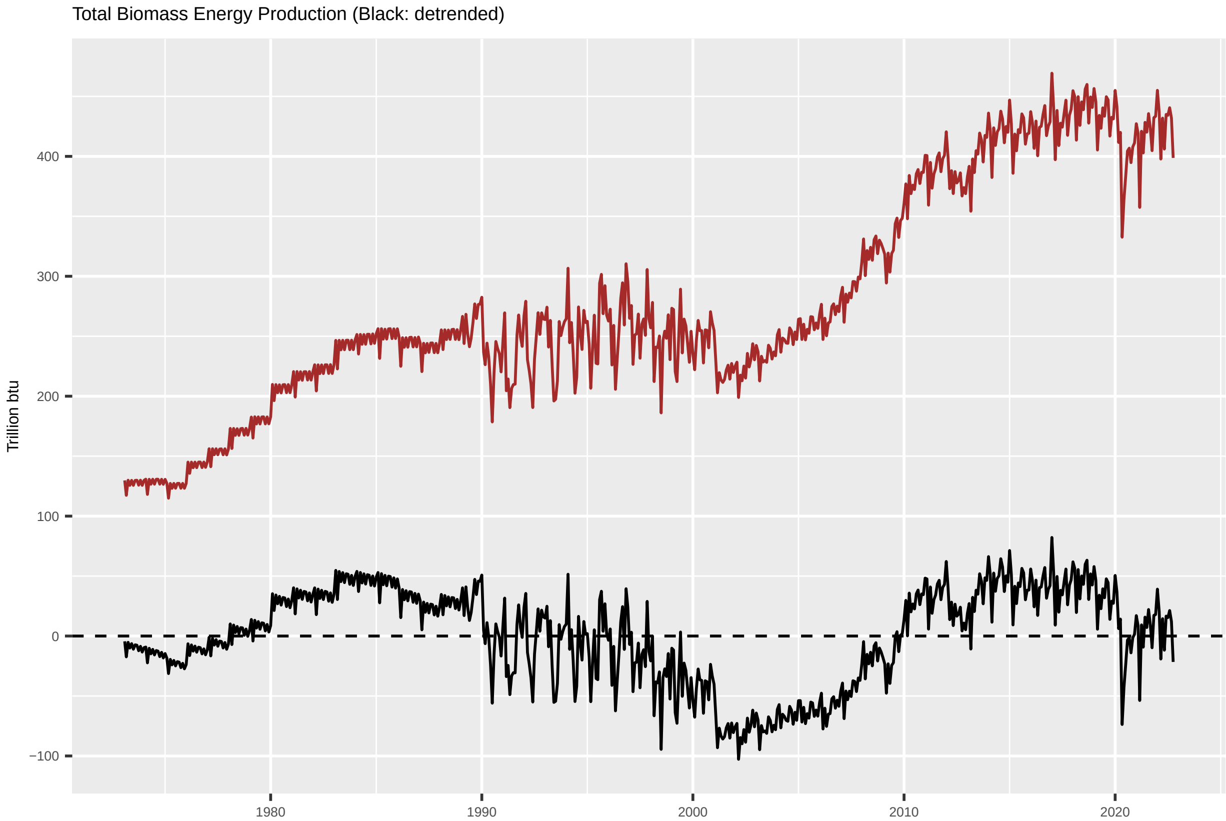
Similarly, the p-value of the remaining two models are both less than 0.001, which indicates significant correlation. The linear trend models can be written as: Renewable energy production = 312 + 0.936*t*
*Hydroelectric power consumption = 260 - 0.0829*t

Q4

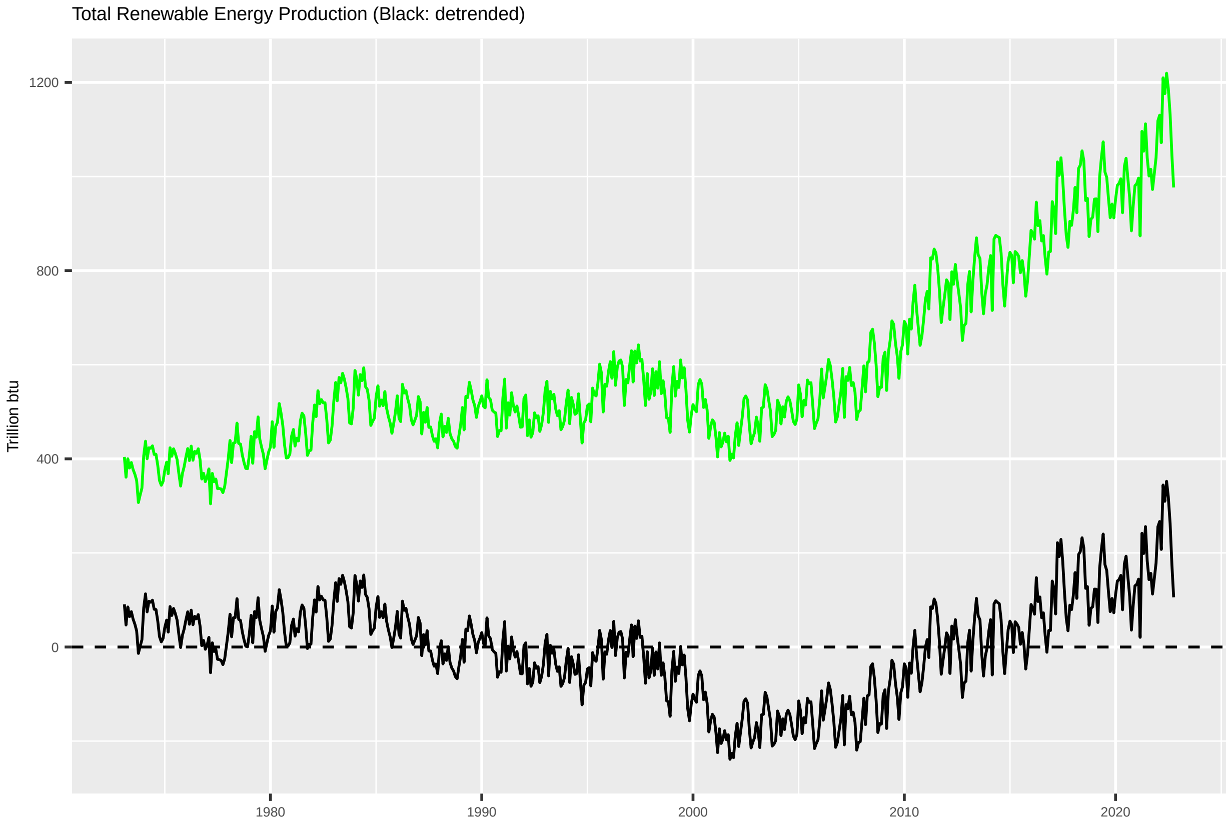
Use the regression coefficients from Q3 to detrend the series. Plot the detrended series and compare with the plots from Q1. What happened? Did anything change?

Answer:

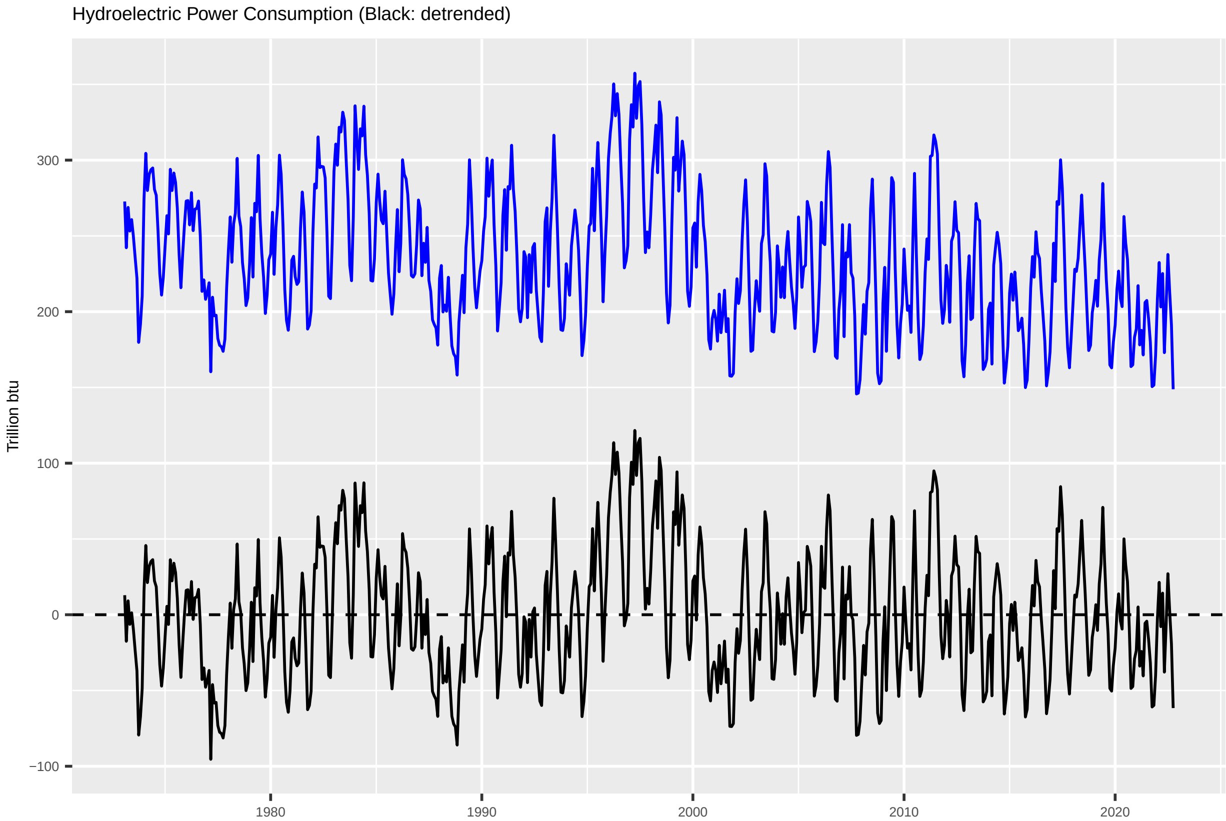
```r
#create an empty list to store plots
plots2_ts <- list()

#plot the de-trended series and compare with the original one
detrend_biomass <- energy_data[,1]-(beta0_biomass + beta1_biomass*t)
ts_detrend_biomass <- ts(detrend_biomass,start = c(1973.1),frequency=12)
print(autoplot(ts_detrend_biomass)+
  labs(title=paste(datanames[1],"(Black: detrended)"),y='Trillion btu')+
  theme(text=element_text(size=6))+
  geom_line(aes(y=ts_energy_data[,1]),color='brown')+
  geom_hline(yintercept = 0, linetype="dashed")
  )
```




```
detrend_renewable <- energy_data[,2]-(beta0_renewable + beta1_renewable*t)
ts_detrend_renewable <- ts(detrend_renewable,start = c(1973.1),frequency=12)
print(autoplot(ts_detrend_renewable)+
  labs(title=paste(datanames[2],"(Black: detrended)"),y='Trillion btu')+
  theme(text=element_text(size=6))+
  geom_line(aes(y=ts_energy_data[,2]),color='green')+
  geom_hline(yintercept = 0, linetype="dashed")
)
```




```r
detrend_hydro <- energy_data[,3]-(beta0_hydro + beta1_hydro*t)
ts_detrend_hydro <- ts(detrend_hydro,start = c(1973.1),frequency=12)
print(autoplot(ts_detrend_hydro)+
  labs(title=paste(datanames[3],"(Black: detrended)"),y='Trillion btu')+
  theme(text=element_text(size=6))+
  geom_line(aes(y=ts_energy_data[,3]),color='blue')+
  geom_hline(yintercept = 0, linetype="dashed"))
```

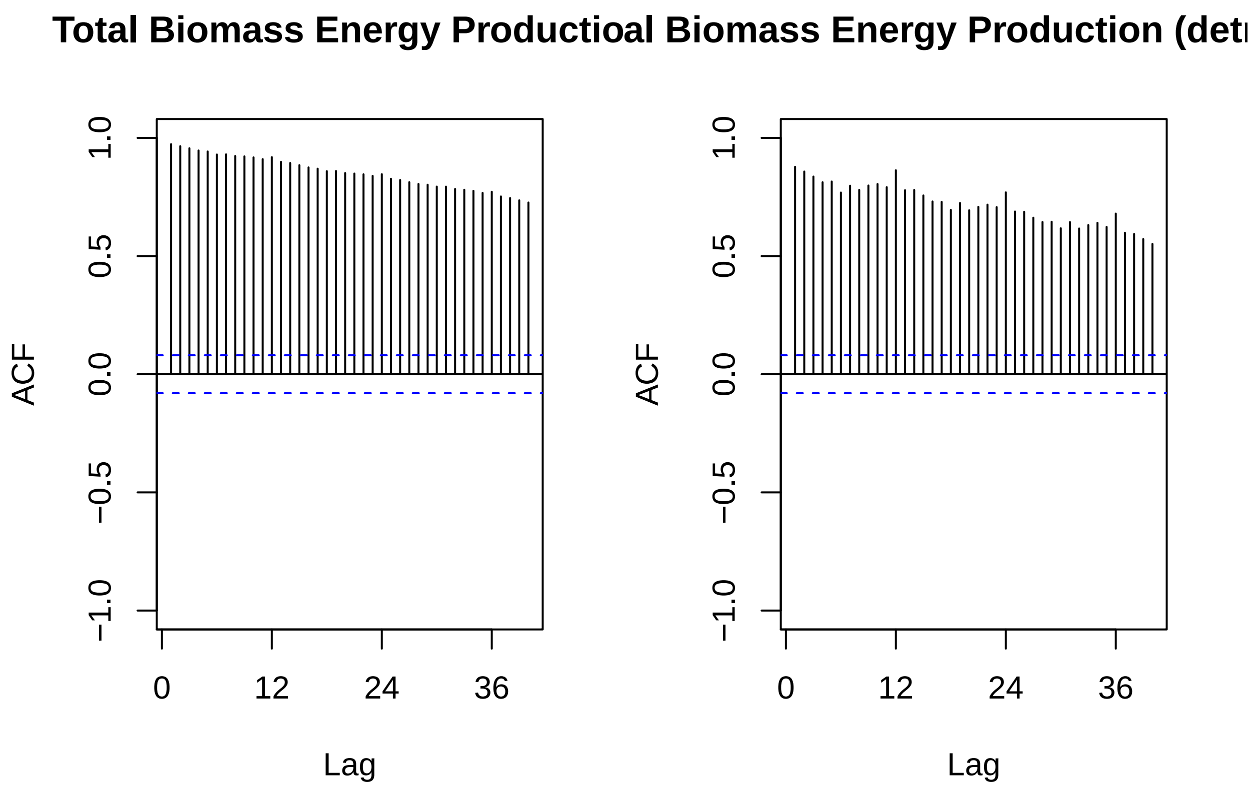


All of the plot changed after being detrended. Firstly, the values of y-axis all decreased a lot, and the detrended lines all flucuate around 0. Also, in terms of the shape of detrended lines, the first half of biomass's line seems to be "lifted up" a bit, making the overall trend more horizontal. Same for renewable energy production. Hydro power consumption changed in the opposite, but the overall trend also get more horizontal.

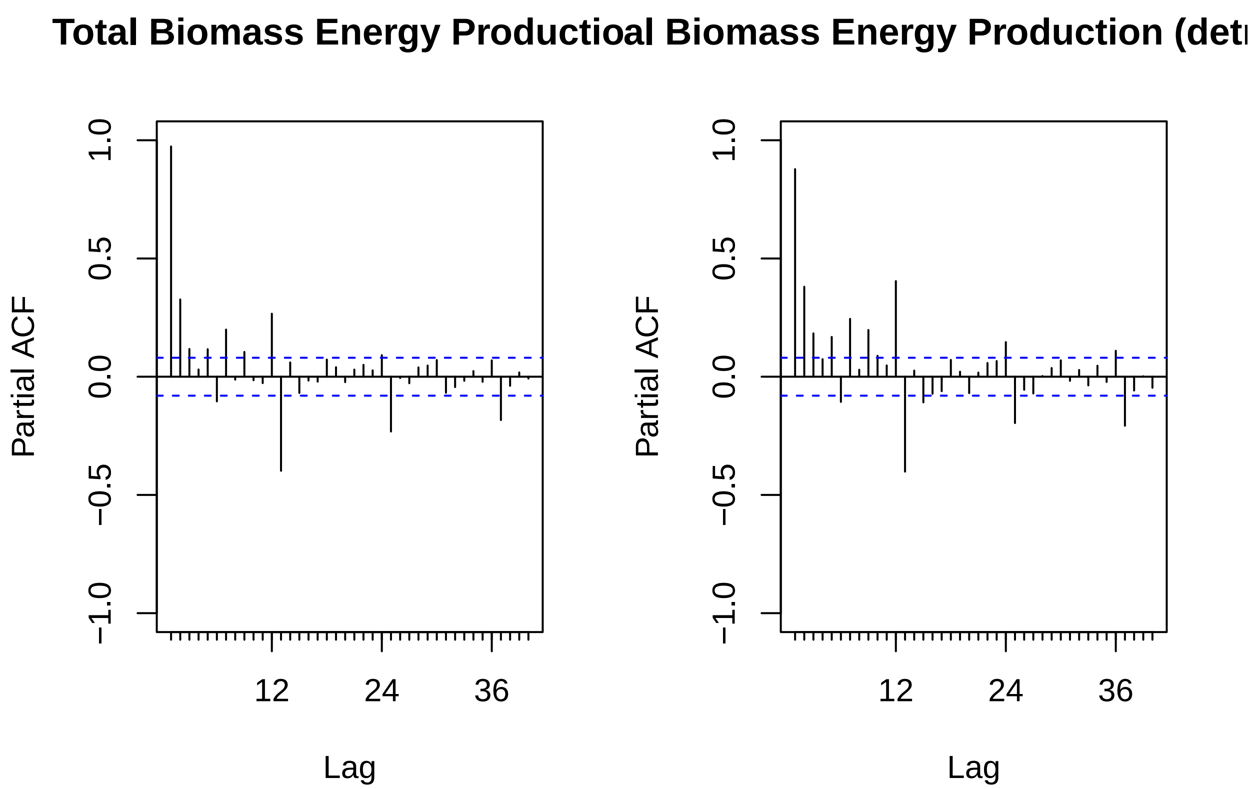
Q5

Plot ACF and PACF for the detrended series and compare with the plots from Q1. Did the plots change? How?

```
#Biomass energy production
par(mfrow=c(1,2))
Acf(ts_energy_data[,1],lag.max=40,main=datanames[1],cex.main=0.3,ylim=c(-1,1))
Acf(ts_detrend_biomass,lag.max = 40, main=paste(datanames[1],"(detrended)"),ylim=c(-1,1),cex.main=0.3, 
```

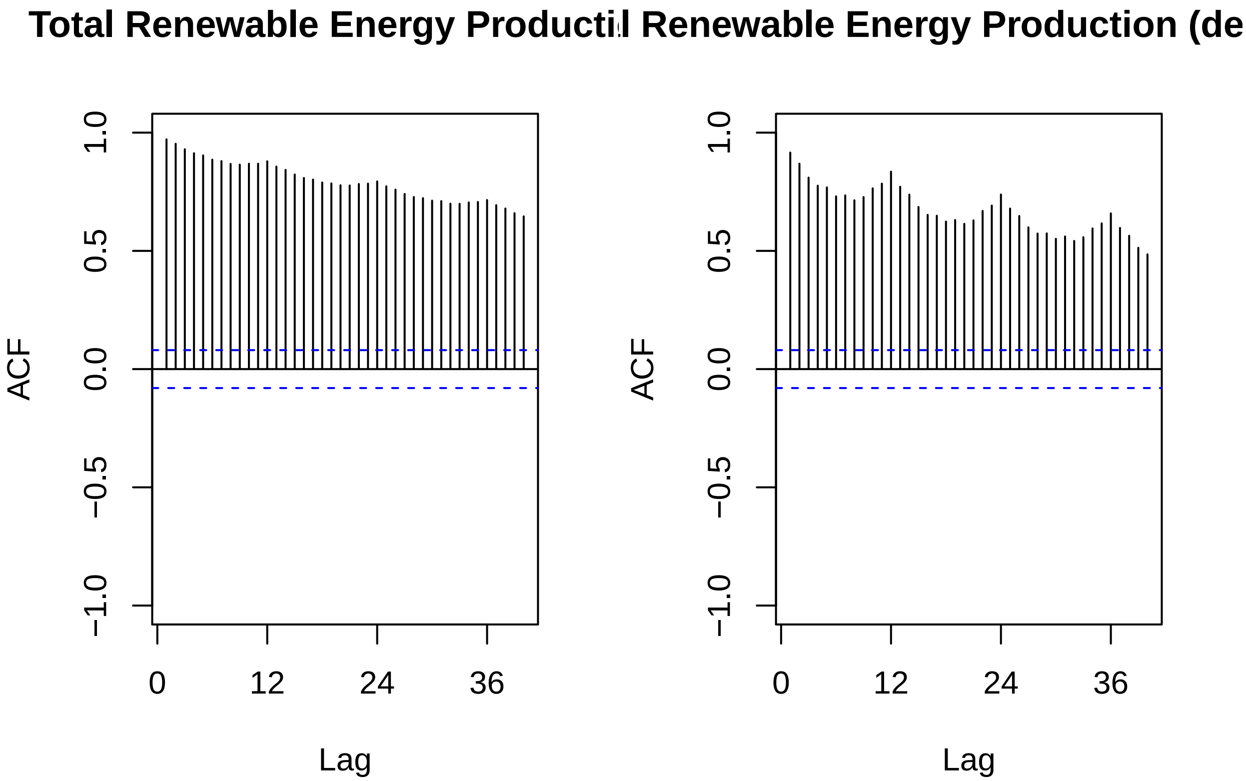



```
Pacf(ts_energy_data[,1],lag.max=40,main=datanames[1],cex.main=0.3,ylim=c(-1,1))
Pacf(ts_detrend_biomass,lag.max = 40, main=paste(datanames[1],"(detrended)"),ylim=c(-1,1),cex.main=0.3)
```

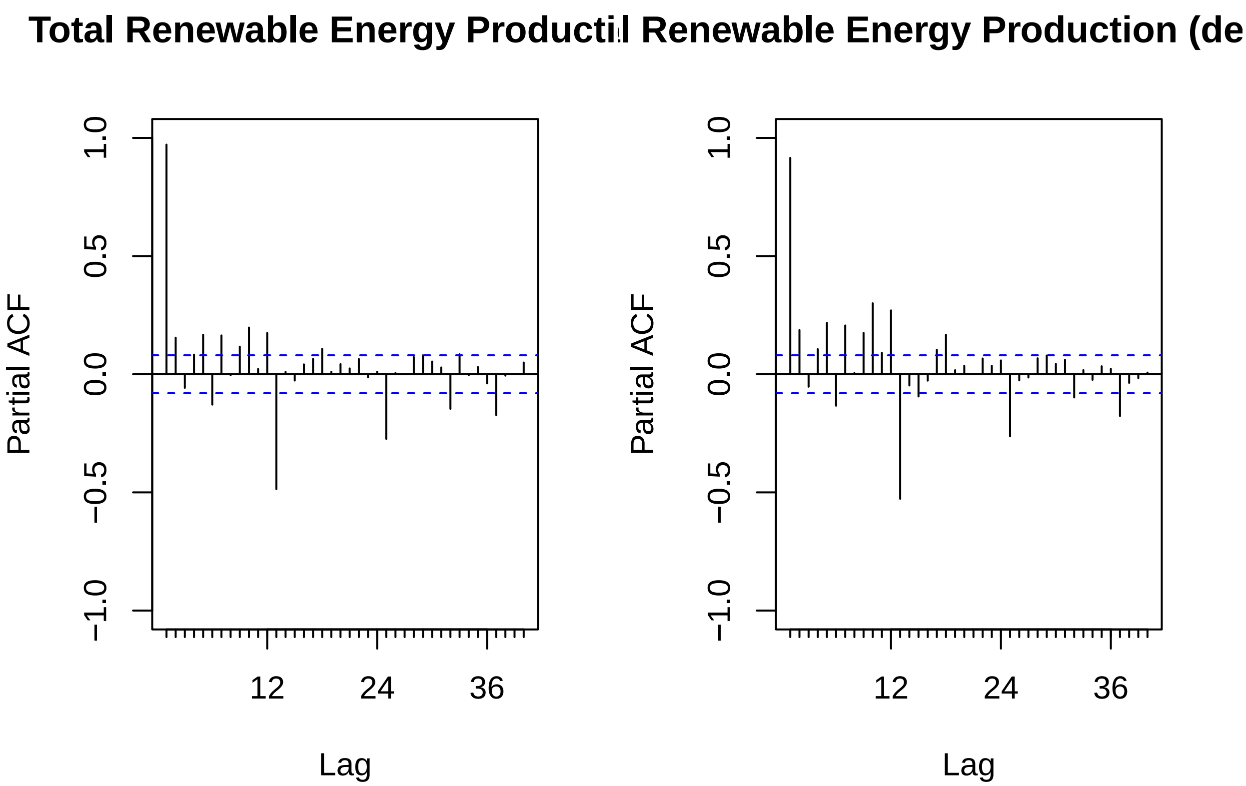


After detrending, most of the values in the ACF plot decreased, except for the values around lag=12, 24, and 36. Likewise, in the PACF plot, only the values around lag=12, 24, and 36 increased.

```
#Renewable energy production
par(mfrow=c(1,2))
Acf(ts_energy_data[,2],lag.max=40,main=datanames[2],cex.main=0.3,ylim=c(-1,1))
Acf(ts_detrend_renewable,lag.max = 40, main=paste(datanames[2],"(detrended)"),ylim=c(-1,1),cex.main=0.3
```

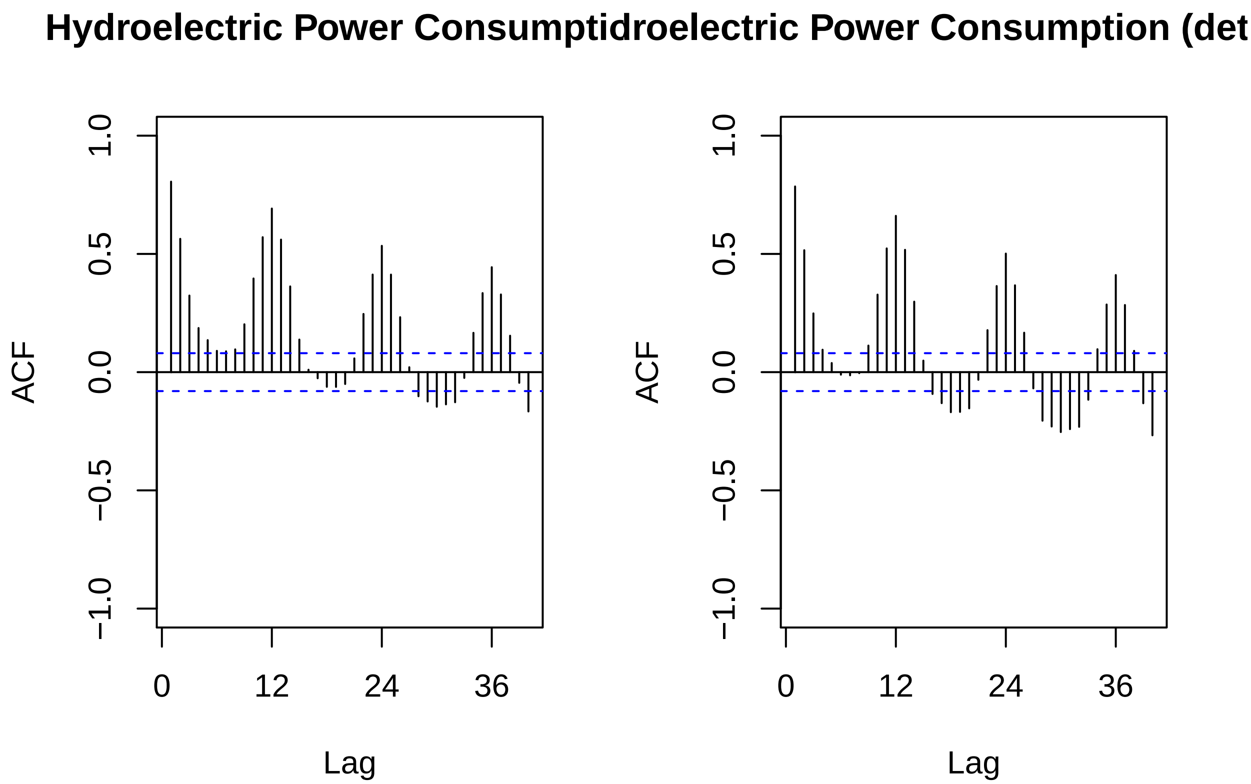



```
Pacf(ts_energy_data[,2],lag.max=40,main=datanames[2],cex.main=0.3,ylim=c(-1,1))
Pacf(ts_detrend_renewable,lag.max = 40, main=paste(datanames[2],"(detrended)"),ylim=c(-1,1),cex.main=0.3
```

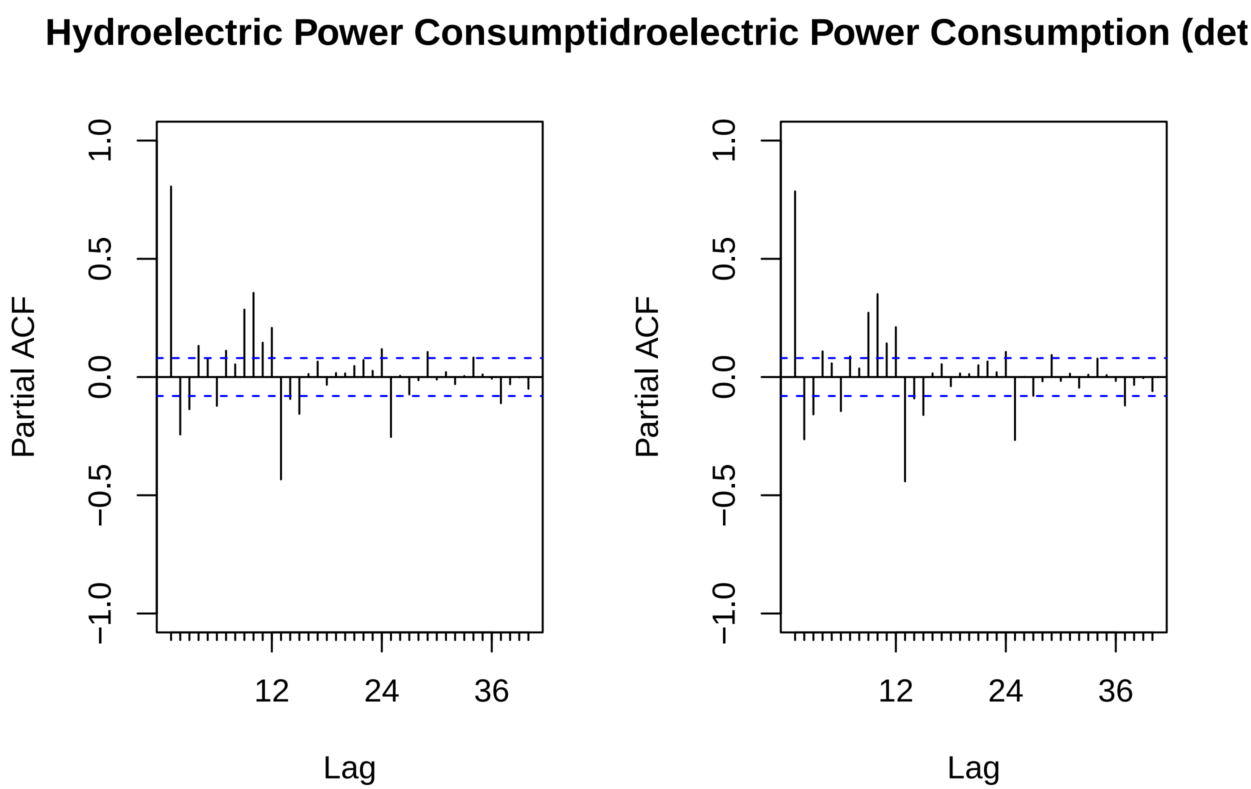


Most of the values in the ACF plot have decreased after the detrending, and the wave-like shape is more pronounced. On the other hand, the values in the PACF plot increased after detrending, indicating a stronger autocorrelation.

```
#Hydro energy consumption
par(mfrow=c(1,2))
Acf(ts_energy_data[,3],lag.max=40,main=datanames[3],cex.main=0.3,ylim=c(-1,1))
Acf(ts_detrend_hydro,lag.max = 40, main=paste(datanames[3],"(detrended)"),ylim=c(-1,1),cex.main=0.3, )
```




```
Pacf(ts_energy_data[,3],lag.max=40,main=datanames[3],cex.main=0.3,ylim=c(-1,1))
Pacf(ts_detrend_hydro,lag.max = 40, main=paste(datanames[3],"(detrended)"),ylim=c(-1,1),cex.main=0.3)
```



The ACF plot shows a stronger, clearer seasonal component after detrending, with deeper troughs and narrower waves. The PACF plot shows no clear change.

Seasonal Component

Set aside the detrended series and consider the original series again from Q1 to answer Q6 to Q8.

Q6

Do the series seem to have a seasonal trend? Which serie/series? Use function *lm()* to fit a seasonal means model (i.e. using the seasonal dummies) to this/these time series. Ask R to print the summary of the regression. Interpret the regression output. Save the regression coefficients for further analysis.

```
dummies <- seasonaldummy(ts_energy_data[,1])

#Biomass energy production
seas_means_model_biomass <- lm(ts_energy_data[,1]~dummies)
summary(seas_means_model_biomass)
```

```
##
## Call:
## lm(formula = ts_energy_data[, 1] ~ dummies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -160.74  -53.67  -24.36   90.73  181.34
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  288.020     13.163  21.881   <2e-16 ***
## dummiesJan    -1.793     18.522  -0.097   0.9229
## dummiesFeb   -31.102     18.522  -1.679   0.0936 .
## dummiesMar    -9.104     18.522  -0.492   0.6232
## dummiesApr   -21.502     18.522  -1.161   0.2462
## dummiesMay   -14.238     18.522  -0.769   0.4424
## dummiesJun   -19.602     18.522  -1.058   0.2904
## dummiesJul    -3.674     18.522  -0.198   0.8428
## dummiesAug    -0.612     18.522  -0.033   0.9737
## dummiesSep   -13.335     18.522  -0.720   0.4718
## dummiesOct    -4.030     18.615  -0.216   0.8287
## dummiesNov    -9.849     18.615  -0.529   0.5970
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 92.14 on 585 degrees of freedom
## Multiple R-squared:  0.01018,    Adjusted R-squared:  -0.008437
## F-statistic: 0.5467 on 11 and 585 DF,  p-value: 0.8714
```

```
beta_int_biomass=seas_means_model_biomass$coefficients[1]
beta_coeff_biomass=seas_means_model_biomass$coefficients[2:12]

#Renewable energy production
seas_means_model_renewable <- lm(ts_energy_data[,2]~dummies)
summary(seas_means_model_renewable)
```

```
##
## Call:
## lm(formula = ts_energy_data[, 2] ~ dummies)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -284.92 -122.23  -68.42   91.22  585.68
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  601.022     27.260  22.048   <2e-16 ***
## dummiesJan    11.468     38.358   0.299    0.765
## dummiesFeb   -41.456     38.358  -1.081    0.280
## dummiesMar    23.130     38.358   0.603    0.547
## dummiesApr     9.959     38.358   0.260    0.795
## dummiesMay    38.853     38.358   1.013    0.312
## dummiesJun    20.378     38.358   0.531    0.595
## dummiesJul     8.298     38.358   0.216    0.829
## dummiesAug   -19.450     38.358  -0.507    0.612
## dummiesSep   -63.770     38.358  -1.662    0.097 .
## dummiesOct   -52.612     38.551  -1.365    0.173
## dummiesNov   -42.537     38.551  -1.103    0.270
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 190.8 on 585 degrees of freedom
## Multiple R-squared:  0.02844,    Adjusted R-squared:  0.01017
## F-statistic: 1.557 on 11 and 585 DF,  p-value: 0.1076
```

```r
beta_int_renewable=seas_means_model_renewable$coefficients[1]
beta_coeff_renewable=seas_means_model_renewable$coefficients[2:12]

#hydroelectric power consumption
seas_means_model_hydro <- lm(ts_energy_data[,3]~dummies)
summary(seas_means_model_hydro)
```

```
##
## Call:
## lm(formula = ts_energy_data[, 3] ~ dummies)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -88.99 -23.47  -2.81  21.99 100.18
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  237.225      4.878  48.634  < 2e-16 ***
## dummiesJan    13.594      6.864   1.981  0.04811 *
## dummiesFeb    -8.254      6.864  -1.203  0.22964
## dummiesMar    19.980      6.864   2.911  0.00374 **
## dummiesApr    15.649      6.864   2.280  0.02297 *
## dummiesMay    39.210      6.864   5.713 1.77e-08 ***
## dummiesJun    31.209      6.864   4.547 6.61e-06 ***
## dummiesJul    10.436      6.864   1.520  0.12895
## dummiesAug   -17.909      6.864  -2.609  0.00931 **
## dummiesSep   -50.173      6.864  -7.310 8.82e-13 ***
## dummiesOct   -48.262      6.898  -6.996 7.22e-12 ***
## dummiesNov   -32.285      6.898  -4.680 3.56e-06 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.14 on 585 degrees of freedom
## Multiple R-squared:  0.4132, Adjusted R-squared:  0.4022
## F-statistic: 37.45 on 11 and 585 DF,  p-value: < 2.2e-16
```

```
beta_int_hydro=seas_means_model_hydro$coefficients[1]
beta_coeff_hydro=seas_means_model_hydro$coefficients[2:12]
```

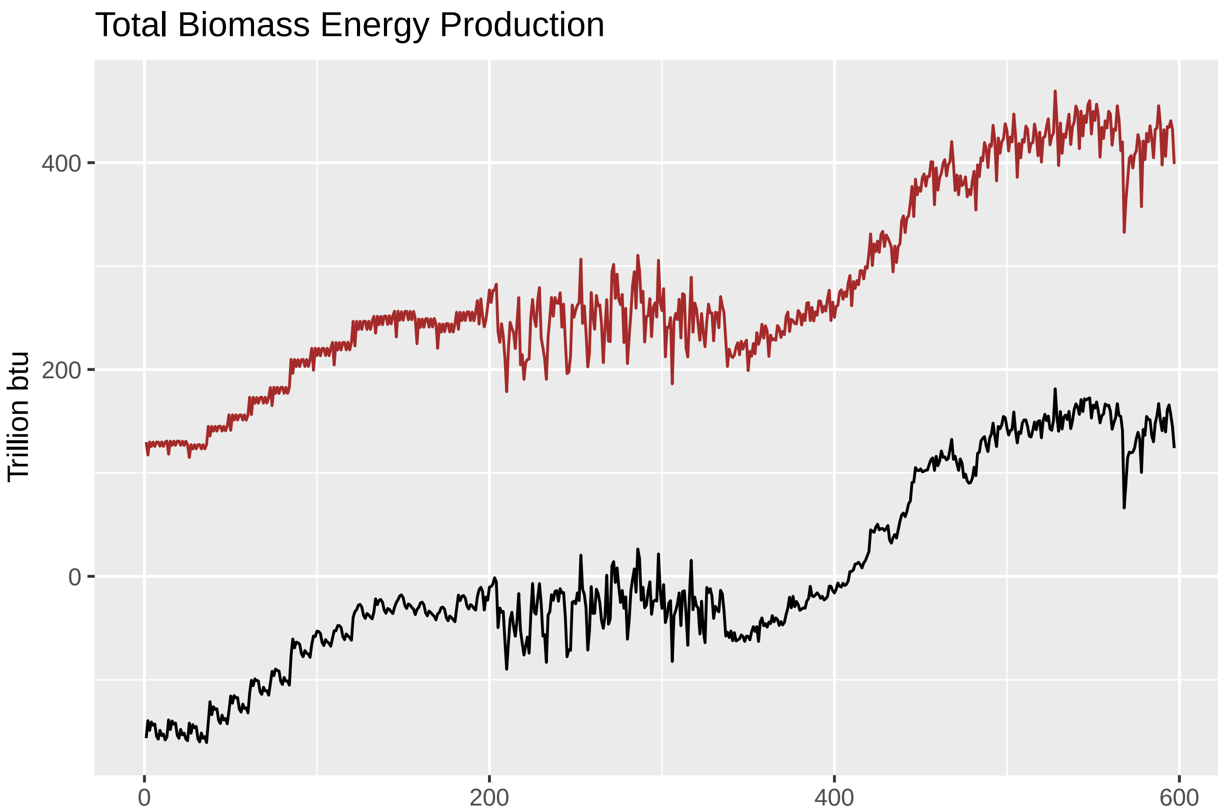
According to the regression output, the p-values in the first two seasonal means models are all greater than 0.05. This indicates that the seasonal means model does not fit the seasonal components well. On the other hand, the seasonal means model works well for the time series of hydroelectric power consumption. Most of the p-values are less than 0.05. The intercept shows the seasonal mean for the month of December, while the other coefficients show the relative change to the seasonal mean of December.

Q7

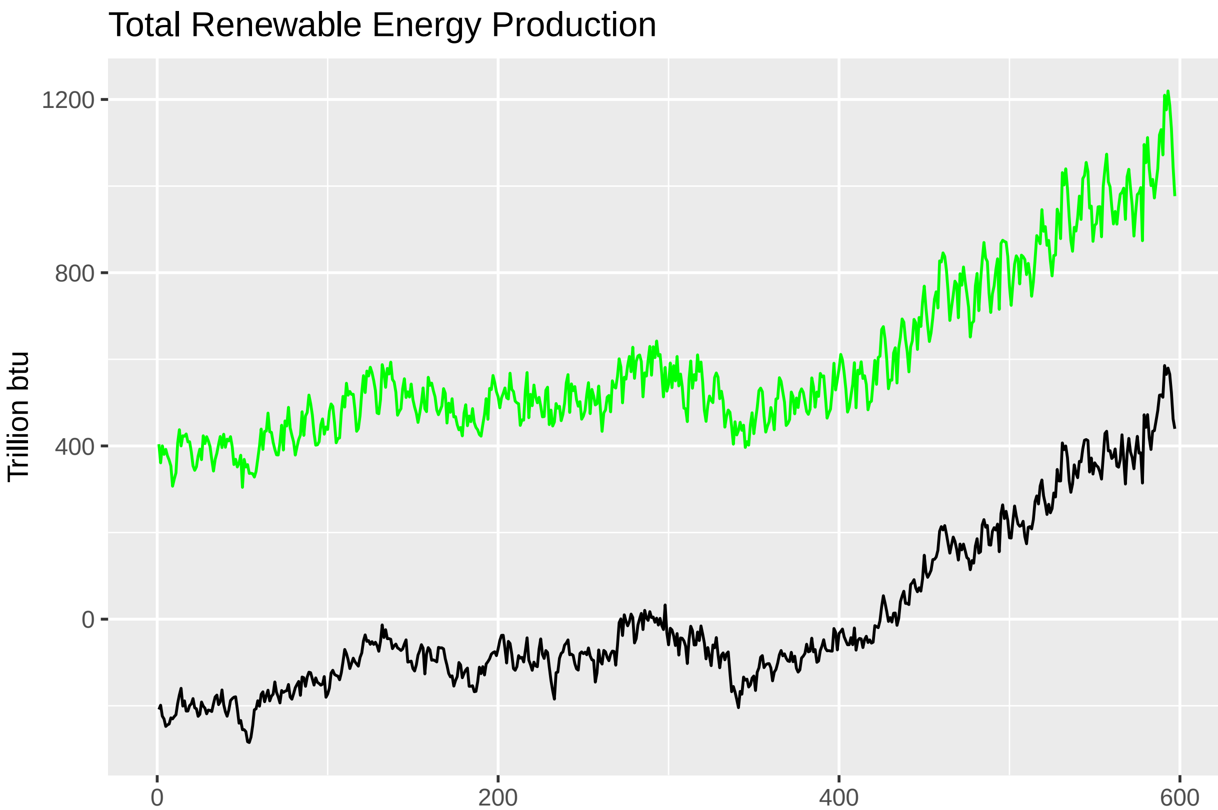
Use the regression coefficients from Q6 to deseason the series. Plot the deseason series and compare with the plots from part Q1. Did anything change?

Answer:

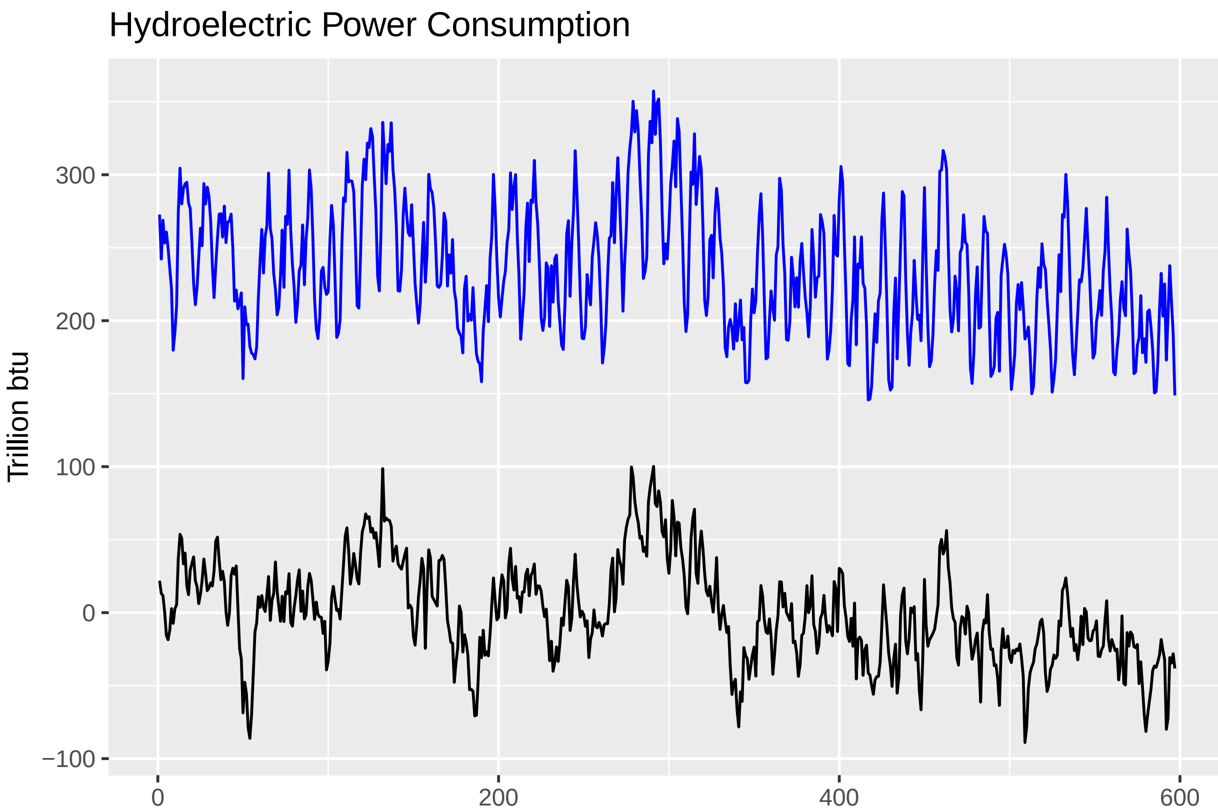
```
#plot biomass energy production with its seasonal component
biomass_seas_comp=array(0,nobs)
for(i in 1:nobs){
  biomass_seas_comp[i]=(beta_int_biomass+beta_coeff_biomass%*%dummies[i,])
}
deseason_biomass <- energy_data[,1] - biomass_seas_comp
ts_deseason_biomass <- ts(deseason_biomass)
print(
  autoplot(ts_deseason_biomass)+
    labs(y='Trillion btu', title = datanames[1])+
    geom_line(aes(y=ts_energy_data[,1]),color='brown')
)
```




```
#same for renewable and hydro
renewable_seas_comp=array(0,nobs)
for(i in 1:nobs){
  renewable_seas_comp[i]=(beta_int_renewable+beta_coeff_renewable%*%dummies[i,])
}
deseason_renewable <- energy_data[,2] - renewable_seas_comp
ts_deseason_renewable <- ts(deseason_renewable)
print(
  autoplot(ts_deseason_renewable)+
    labs(y='Trillion btu', title = datanames[2])+
    geom_line(aes(y=ts_energy_data[,2]),color='green')
)
```




```
hydro_seas_comp=array(0,nobs)
for(i in 1:nobs){
  hydro_seas_comp[i]=(beta_int_hydro+beta_coeff_hydro%*%dummies[i,])
}
deseason_hydro <- energy_data[,3] - hydro_seas_comp
ts_deseason_hydro <- ts(deseason_hydro)
print(
  autoplot(ts_deseason_hydro)+
    labs(y='Trillion btu', title = datanames[3])+
    geom_line(aes(y=ts_energy_data[,3]),color='blue')
)
```

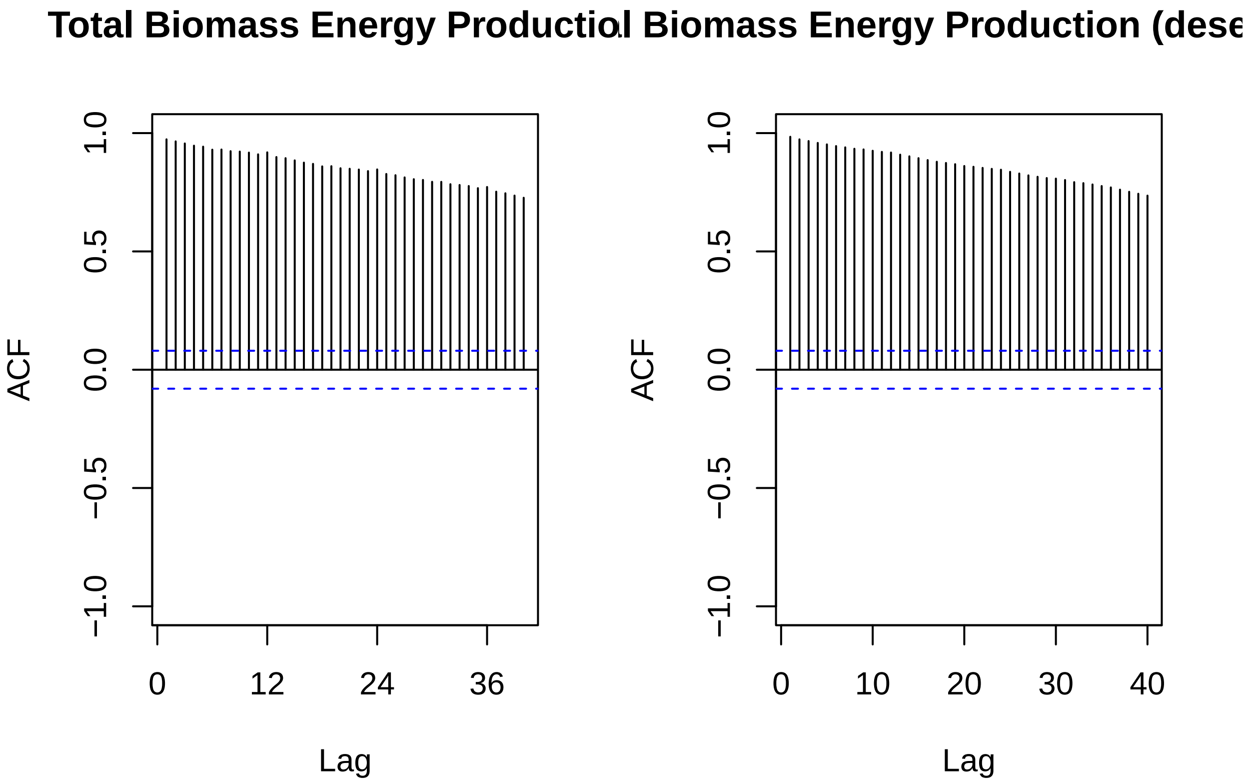


Yes. Not only do the values decrease after deseasoning, but the random fluctuations in each line seem to be more noticeable.

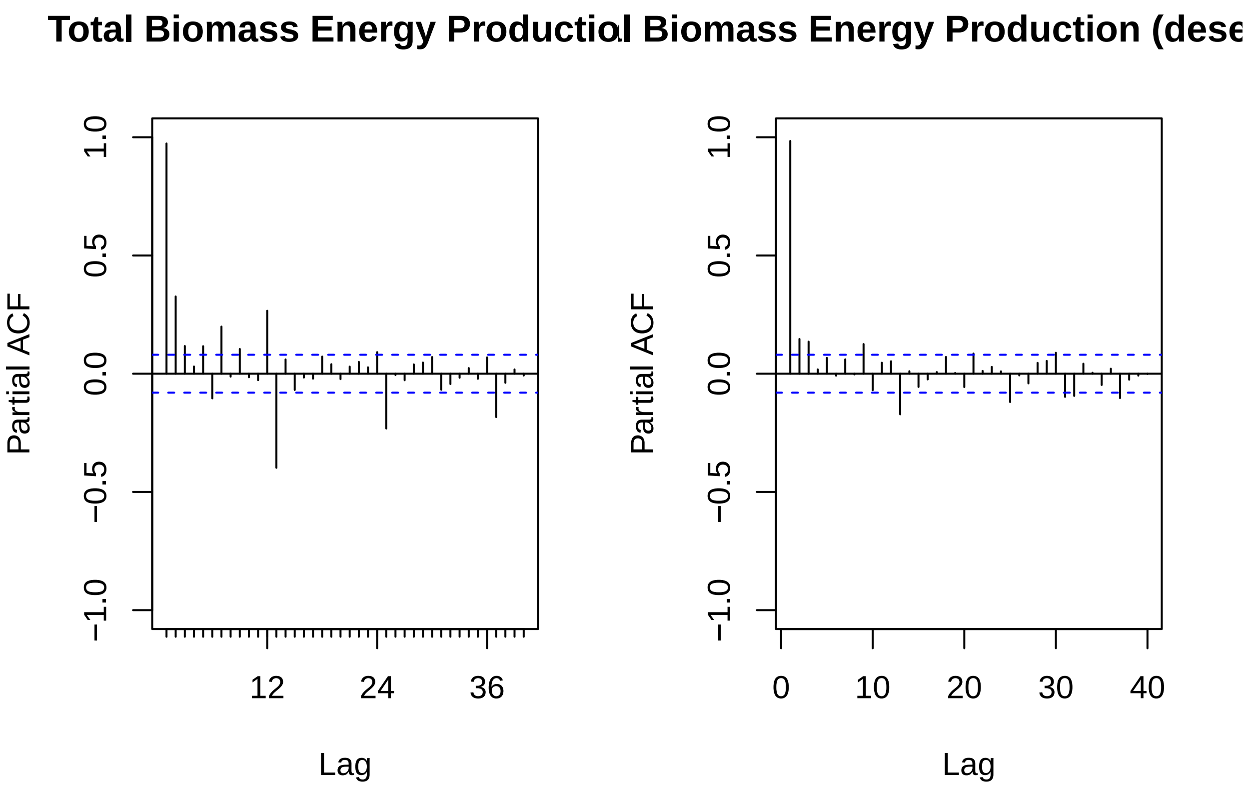
Q8

Plot ACF and PACF for the deseason series and compare with the plots from Q1. Did the plots change? How?

```
#Biomass energy production
par(mfrow=c(1,2))
Acf(ts_energy_data[,1],lag.max=40,main=datanames[1],cex.main=0.3,ylim=c(-1,1))
Acf(ts_deseason_biomass,lag.max = 40, main=paste(datanames[1],"(deseasoned)"),ylim=c(-1,1),cex.main=0.3
```

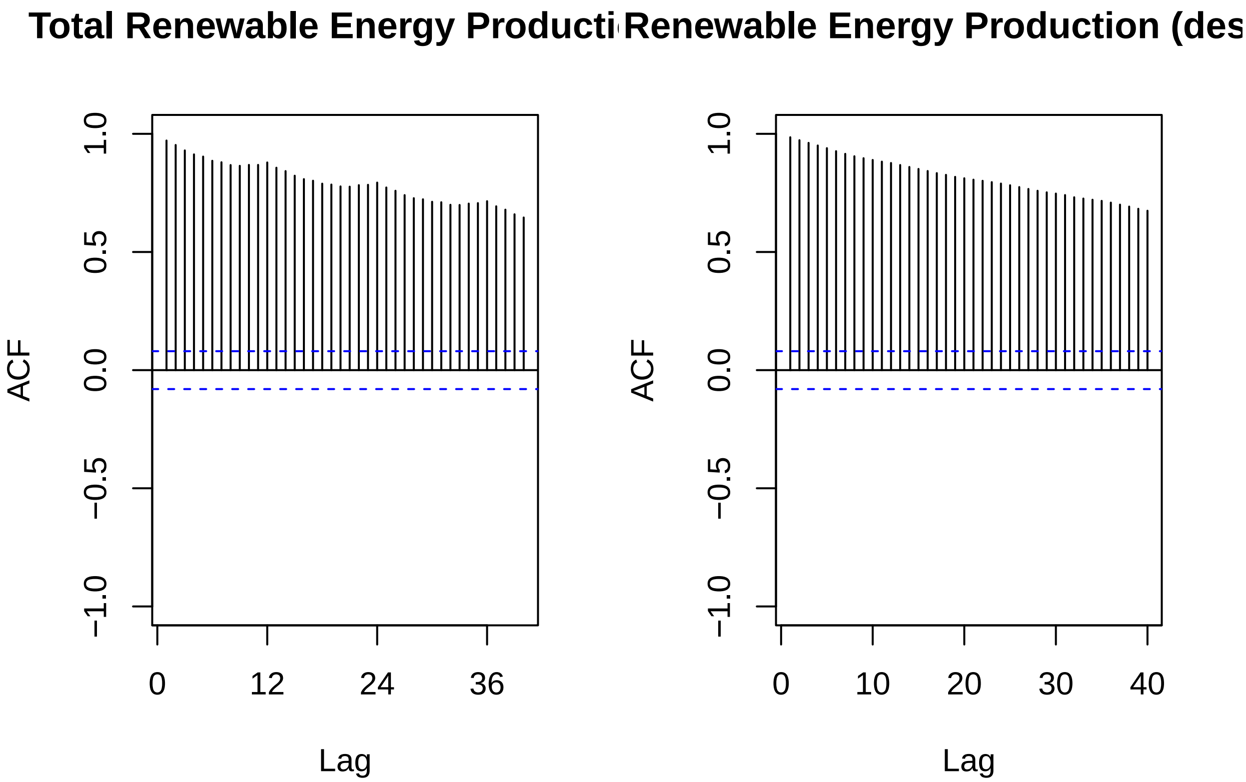



```
Pacf(ts_energy_data[,1],lag.max=40,main=datanames[1],cex.main=0.3,ylim=c(-1,1))
Pacf(ts_deseason_biomass,lag.max = 40, main=paste(datanames[1],"(deseasoned)"),ylim=c(-1,1),cex.main=0.
```

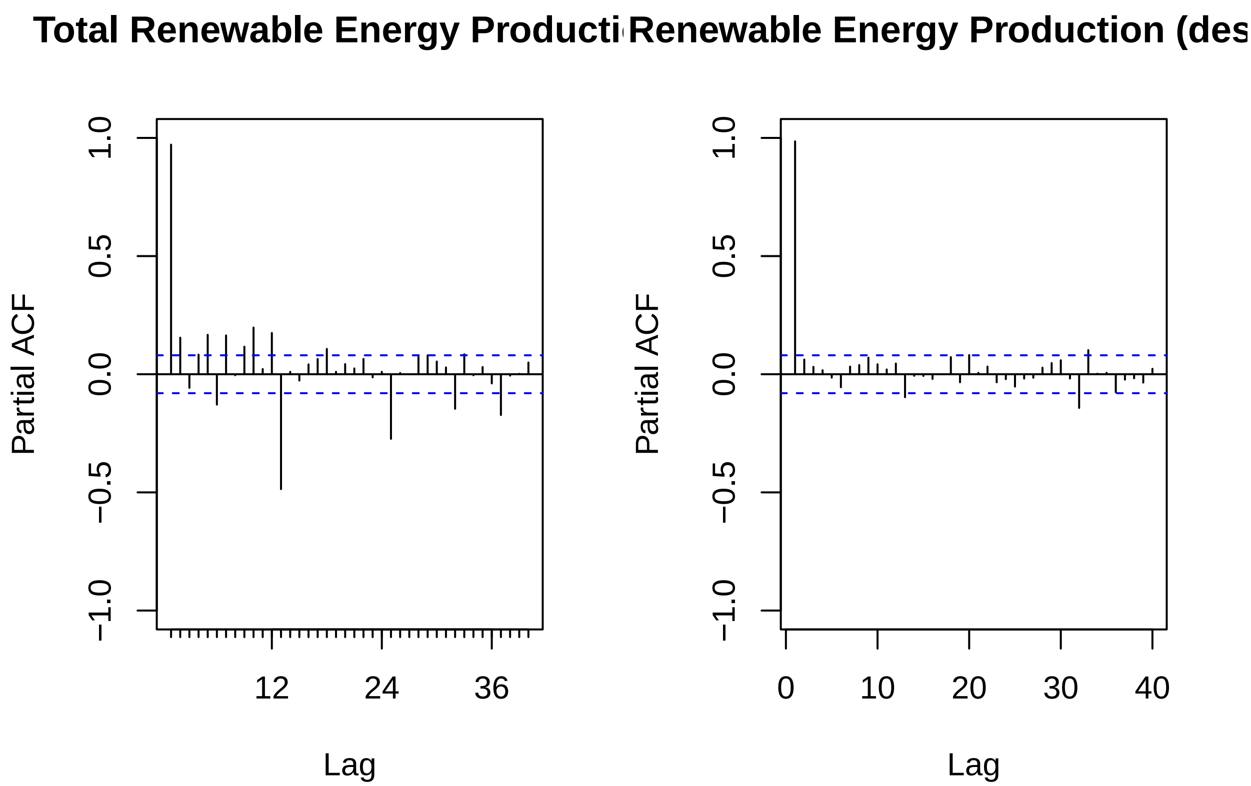



```
#Renewable energy production
par(mfrow=c(1,2))
```

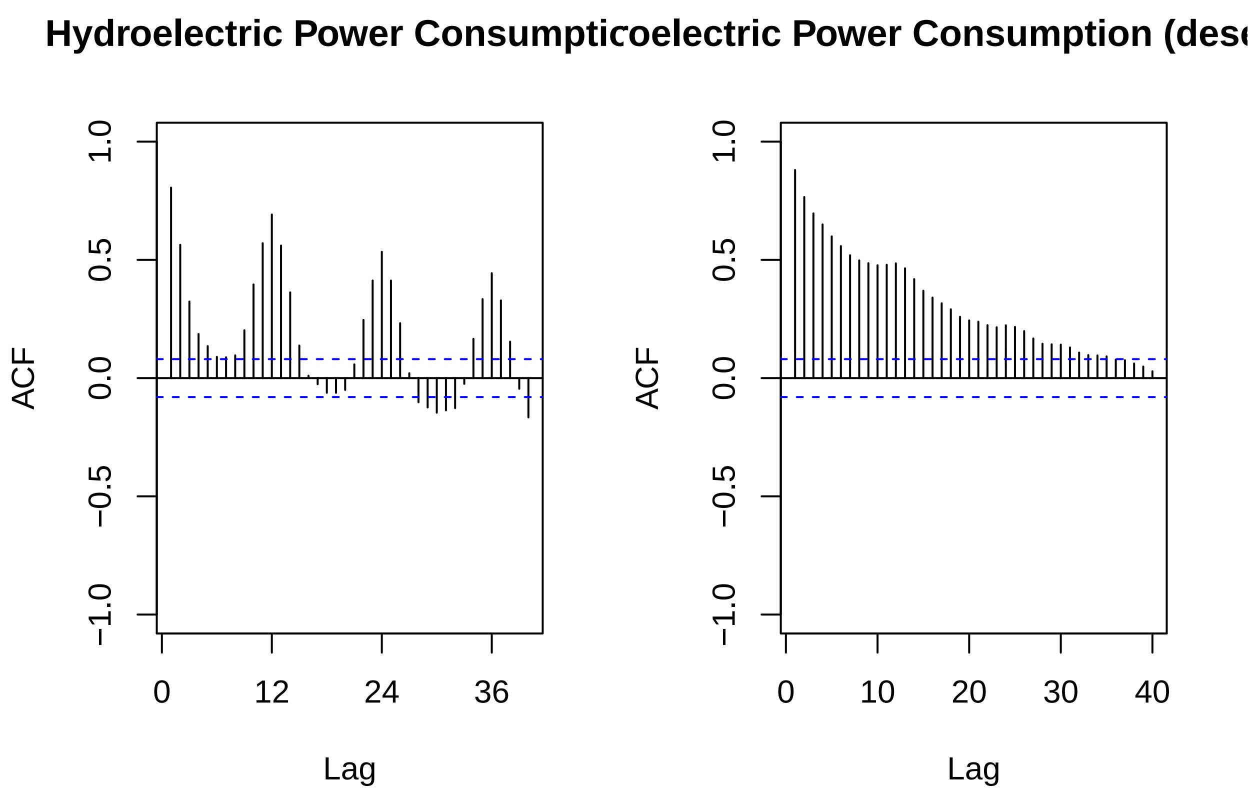
```
Acf(ts_energy_data[,2],lag.max=40,main=datanames[2],cex.main=0.3,ylim=c(-1,1))
Acf(ts_deseason_renewable,lag.max = 40, main=paste(datanames[2],"(deseasoned)"),ylim=c(-1,1),cex.main=0
```



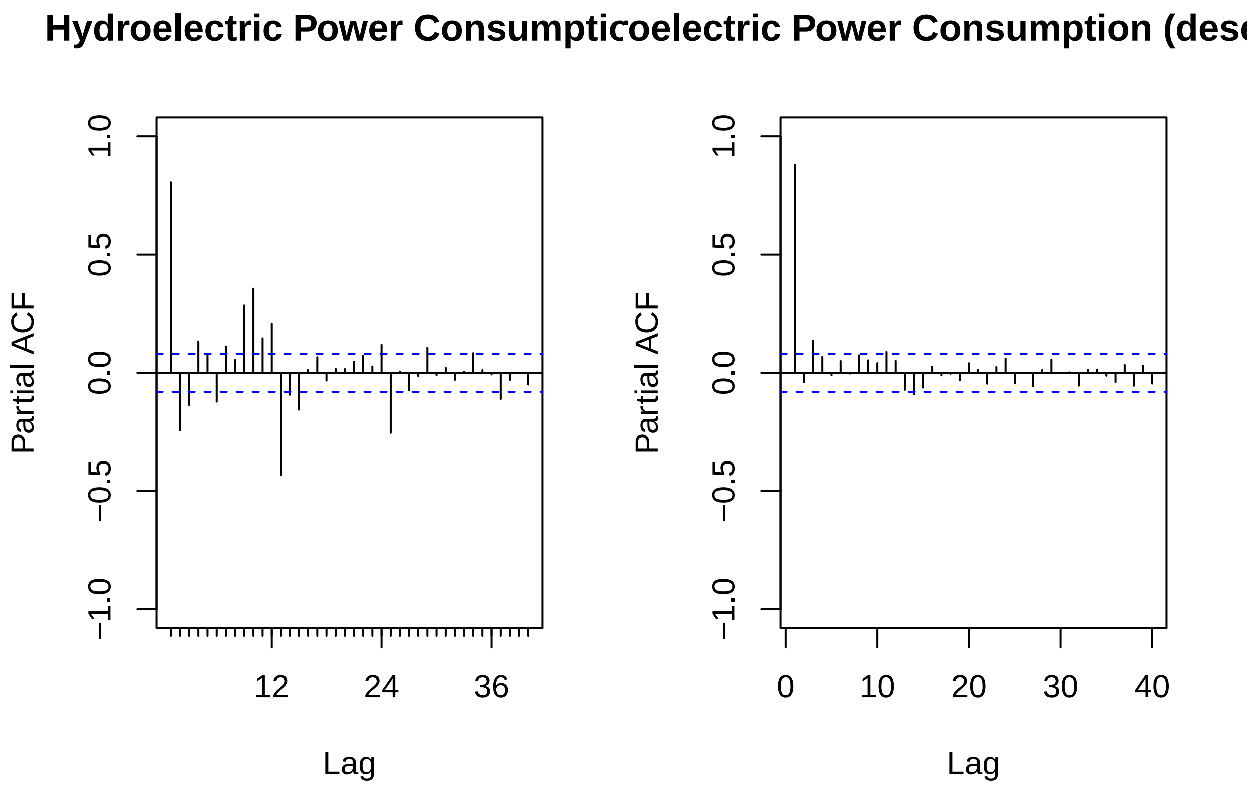

```
Pacf(ts_energy_data[,2],lag.max=40,main=datanames[2],cex.main=0.3,ylim=c(-1,1))
Pacf(ts_deseason_renewable,lag.max = 40, main=paste(datanames[2],"(deseasoned)"),ylim=c(-1,1),cex.main=
```




```
#Hydro energy consumption
par(mfrow=c(1,2))
Acf(ts_energy_data[,3],lag.max=40,main=datanames[3],cex.main=0.3,ylim=c(-1,1))
Acf(ts_deseason_hydro,lag.max = 40, main=paste(datanames[3],"(deseasoned)"),ylim=c(-1,1),cex.main=0.3,
```



```
Pacf(ts_energy_data[,3],lag.max=40,main=datanames[3],cex.main=0.3,ylim=c(-1,1))
Pacf(ts_deseason_hydro,lag.max = 40, main=paste(datanames[3],"(deseasoned)"),ylim=c(-1,1),cex.main=0.3)
```



Yes, all plots changed. Any wave-like patterns in original ACF and PACF plots no longer exist. In addition, the PACF values decay much faster after deseasoning. There are very few values remaining above significant levels, most of which are located at lags close to 0.