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

Assignment 2 - Due date 02/03/23

Qiuying Liao

Submission Instructions

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).

When you have completed the assignment, **Knit** the text and code into a single PDF file. Submit this pdf using Sakai.

R packages

R packages needed for this assignment:"forecast","tseries", and "dplyr". 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(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

Data set information

Consider the data provided in the spreadsheet "Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.x on our **Data** folder. The data comes from the US Energy Information and Administration and corresponds to the December 2022 Monthly Energy Review. The spreadsheet is ready to be used. You will also find a *.csv* version of the data “Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source-Edit.csv”. You may use the function *read.table*() to import the *.csv* data in R. Or refer to the file “M2_ImportingData_CSV_XLSX.Rmd” in our Lessons folder for functions that are better suited for importing the *.xlsx*.

```
#Importing data set using read.csv
energy_data <- read.csv(file="../Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source-
```

Question 1

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. Use the command head() to verify your data.

```
#create data frame
energy_df <- data.frame(energy_data$Month,energy_data$Total.Biomass.Energy.Production, energy_data$Total

head(energy_df)
```

```
##   energy_data.Month energy_data.Total.Biomass.Energy.Production
## 1      1973 January                                     129.787
## 2     1973 February                                     117.338
## 3        1973 March                                     129.938
## 4        1973 April                                     125.636
## 5          1973 May                                     129.834
## 6         1973 June                                     125.611
##   energy_data.Total.Renewable.Energy.Production
## 1                                       403.981
## 2                                       360.900
## 3                                       400.161
## 4                                       380.470
## 5                                       392.141
## 6                                       377.232
##   energy_data.Hydroelectric.Power.Consumption
## 1                                     272.703
## 2                                     242.199
## 3                                     268.810
## 4                                     253.185
## 5                                     260.770
## 6                                     249.859
```

Question 2

Transform your data frame in a time series object and specify the starting point and frequency of the time series using the function ts().

```
#Convert month into format
Date <- paste(energy_df[,1], "01", sep = "")
Date_new <- as.Date (Date, format="%Y %B %d")
head(Date_new)
```

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

```
energy_df <- cbind(Date_new, energy_df[,2:4])
head(energy_df)
```

```
##     Date_new energy_data.Total.Biomass.Energy.Production
## 1 1973-01-01                                     129.787
## 2 1973-02-01                                     117.338
## 3 1973-03-01                                     129.938
## 4 1973-04-01                                     125.636
## 5 1973-05-01                                     129.834
## 6 1973-06-01                                     125.611
##   energy_data.Total.Renewable.Energy.Production
## 1                                       403.981
## 2                                       360.900
## 3                                       400.161
## 4                                       380.470
## 5                                       392.141
## 6                                       377.232
##   energy_data.Hydroelectric.Power.Consumption
## 1                                     272.703
## 2                                     242.199
## 3                                     268.810
## 4                                     253.185
## 5                                     260.770
## 6                                     249.859
```

```
#time series
ts_energy_df <- ts(energy_df[,2:4], start = c(1973,1), frequency = 12)
```

Question 3

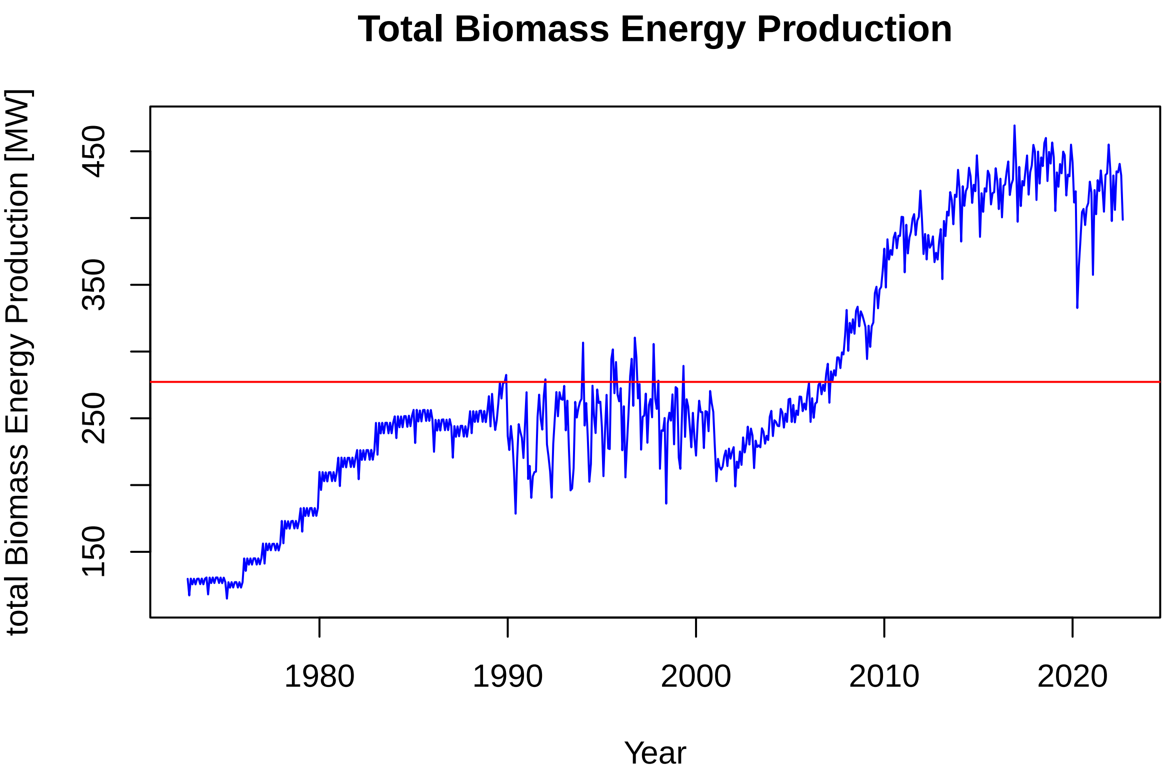
Compute mean and standard deviation for these three series.

```
#rewrite column names
colnames(ts_energy_df)=c("Biomass","Renewable", "Hydroelectric")

mean_biomass <- mean(ts_energy_df[,"Biomass"])
mean_renewable <- mean(ts_energy_df[,"Renewable"])
mean_hydroelectric <- mean(ts_energy_df[,"Hydroelectric"])

sd_biomass <- sd(ts_energy_df[,"Biomass"])
sd_renewable <- sd(ts_energy_df[,"Renewable"])
sd_hydroelectric <- sd(ts_energy_df[,"Hydroelectric"])
```

Question 4

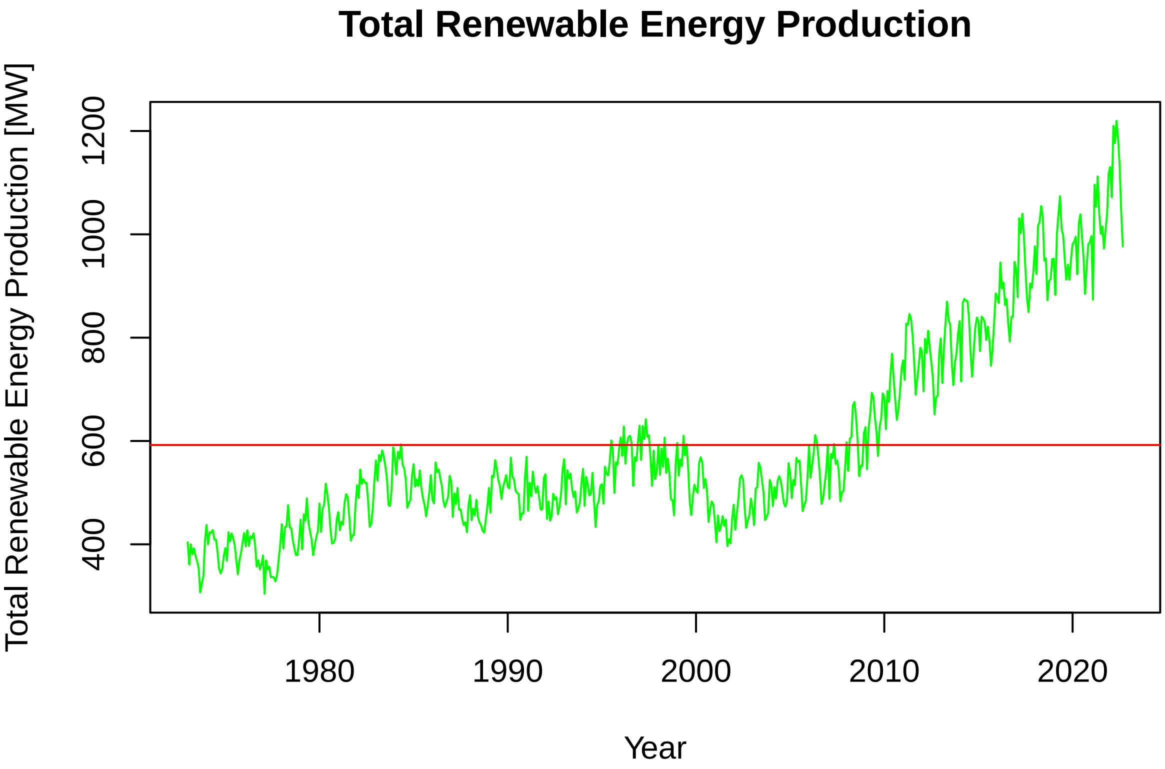
Display and interpret the time series plot for each of these variables. Try to make your plot as informative as possible by writing titles, labels, etc. For each plot add a horizontal line at the mean of each series in a different color.

```
#Graph 1: Plot the series for Total Biomass Energy Production
plot(ts_energy_df[,"Biomass"],type="l",col="blue",ylab="total Biomass Energy Production [MW]", xlab="Yea

#Additional - Suppose you want to add a line with the mean
abline(h=mean(ts_energy_df[,"Biomass"]),col="red")
```



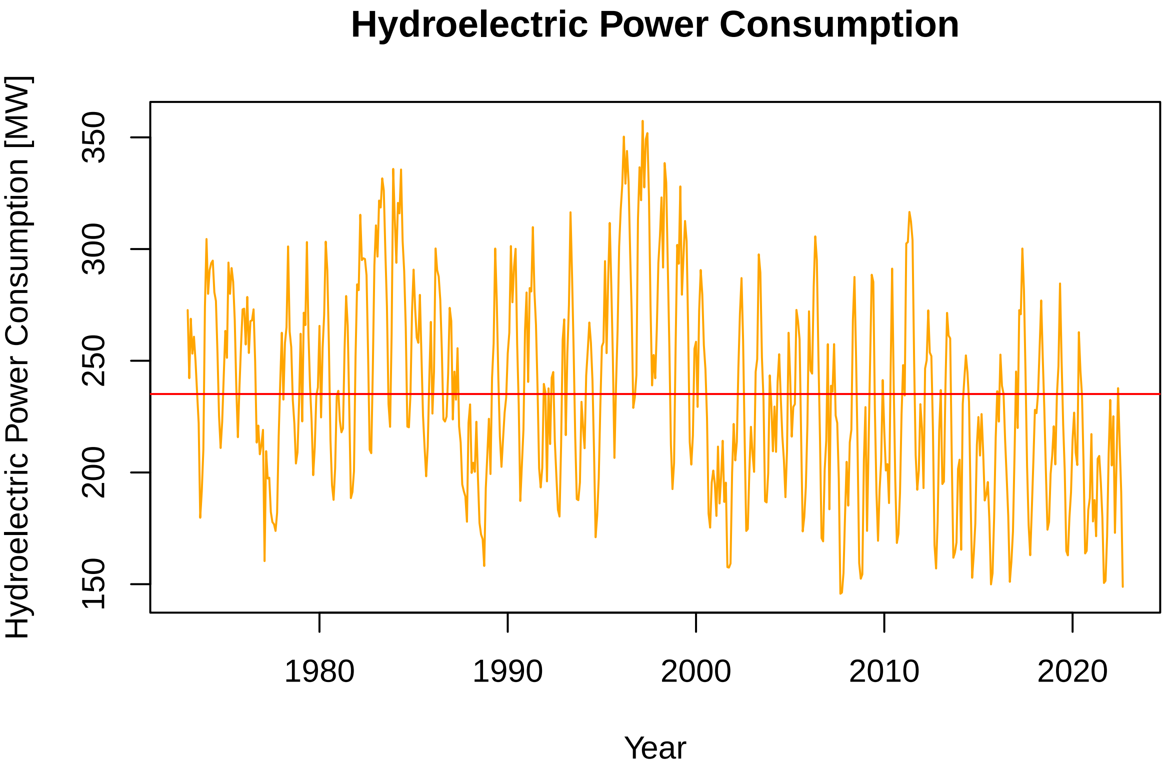
The graph shows a general upward trend in total biomass energy production. total biomass energy production increased from 1973 to 1989, but fluctuated more around the mean from 1990 to 2000 and increased after 2000. there was a sharp decline around 2020. The mean value is about 277 MW with a standard deviation of about 91.8.

```
#Graph 2: Plot the series for Total Renewable Energy Production
plot(ts_energy_df[,"Renewable"],type="l",col="green",ylab="Total Renewable Energy Production [MW]", xlal

#Additional - Suppose you want to add a line with the mean
abline(h=mean(ts_energy_df[,"Renewable"]),col="red")
```



The graph shows a general upward trend in total renewable energy production. It fluctuated below the mean from 1990 to 2010 and increased after 2000 (above mean). The fluctuation is relatively consistent across years. The mean value is about 592.2 MW with a standard deviation of about 191.8.

```
#Graph 3: Plot the series for hydroelectric power consumption
plot(ts_energy_df[,"Hydroelectric"],type="l",col="orange",ylab="Hydroelectric Power Consumption [MW]", 

#Additional - Suppose you want to add a line with the mean
abline(h=mean(ts_energy_df[,"Hydroelectric"]),col="red")
```



The graph shows a general downward trend in total hydroelectric power consumption. It has high fluctuation, fluctuating around the mean value of each year. The mean value is about 235.1 MW with a standard deviation of about 44.2.

Question 5

Compute the correlation between these three series. Are they significantly correlated? Explain your answer.

*Biomass and renewable is significantly and highly positive correlated with 0.92 value. Biomass and hydroelectric is significantly but medium positive correlated with -0.39 value. Renewable and hydroelectric has low negative correlation and may not significant with -0.099 value.**

```
cor(ts_energy_df)
```

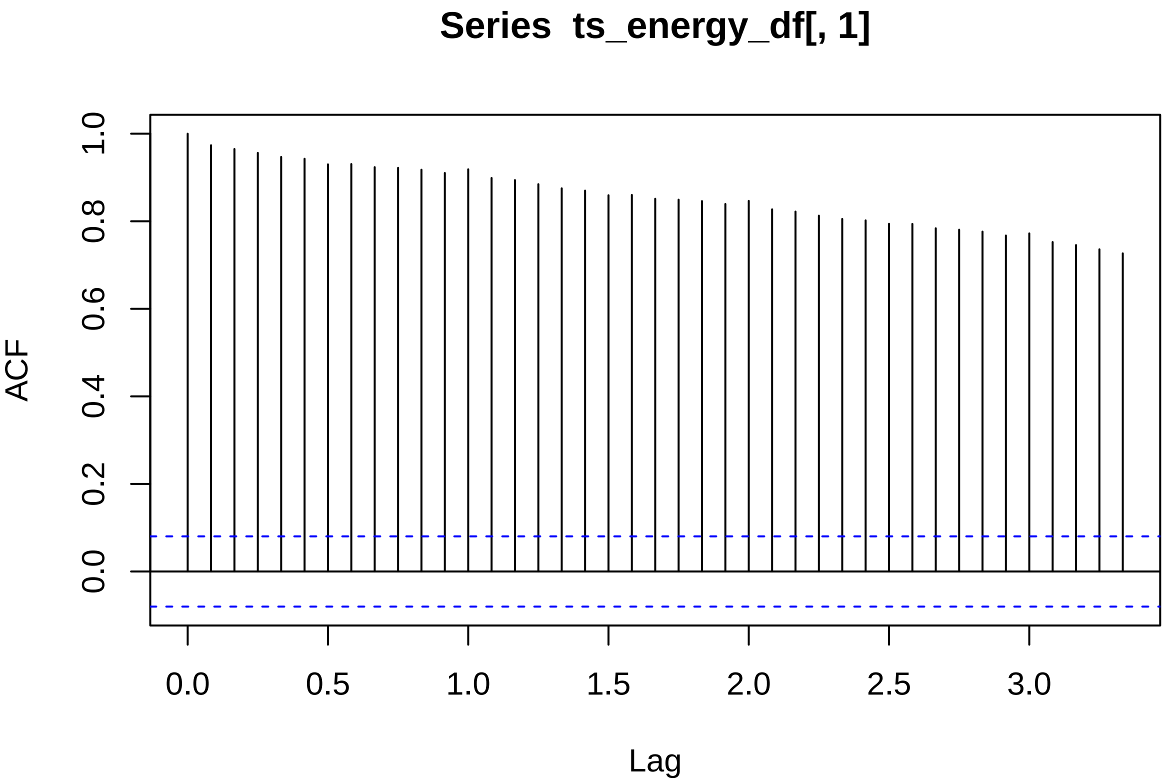
```
##                  Biomass   Renewable Hydroelectric
## Biomass        1.0000000  0.91859411   -0.29982013
## Renewable      0.9185941  1.00000000   -0.09958758
## Hydroelectric -0.2998201 -0.09958758    1.00000000
```

Question 6

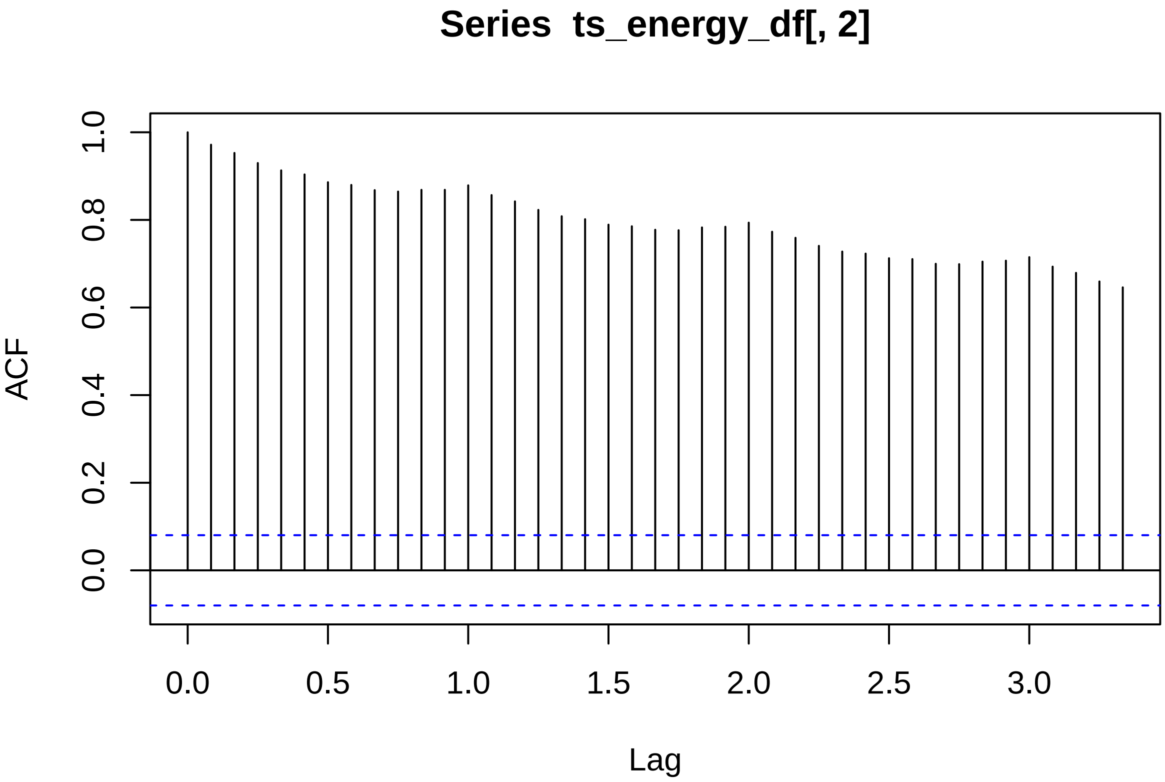
Compute the autocorrelation function from lag 1 up to lag 40 for these three variables. What can you say about these plots? Do the three of them have the same behavior?

The first and second graphs show nonstationarity in the time series. In addition, they are all positive in both graphs. There is no obvious seasonality for both graph. The correlation is high even at lag 3.0. On the other hand, the third graph shows stationarity in the series. The correlation is slightly decreasing over time, and there is a seasonality trend in the third graph. The autocorrelation is still strong at lag 3.0 since the value is above the blue line. Value within blue line shows weak correlation and may not significant. The first two graphs have similar behavior but the third graph shows different behavior.

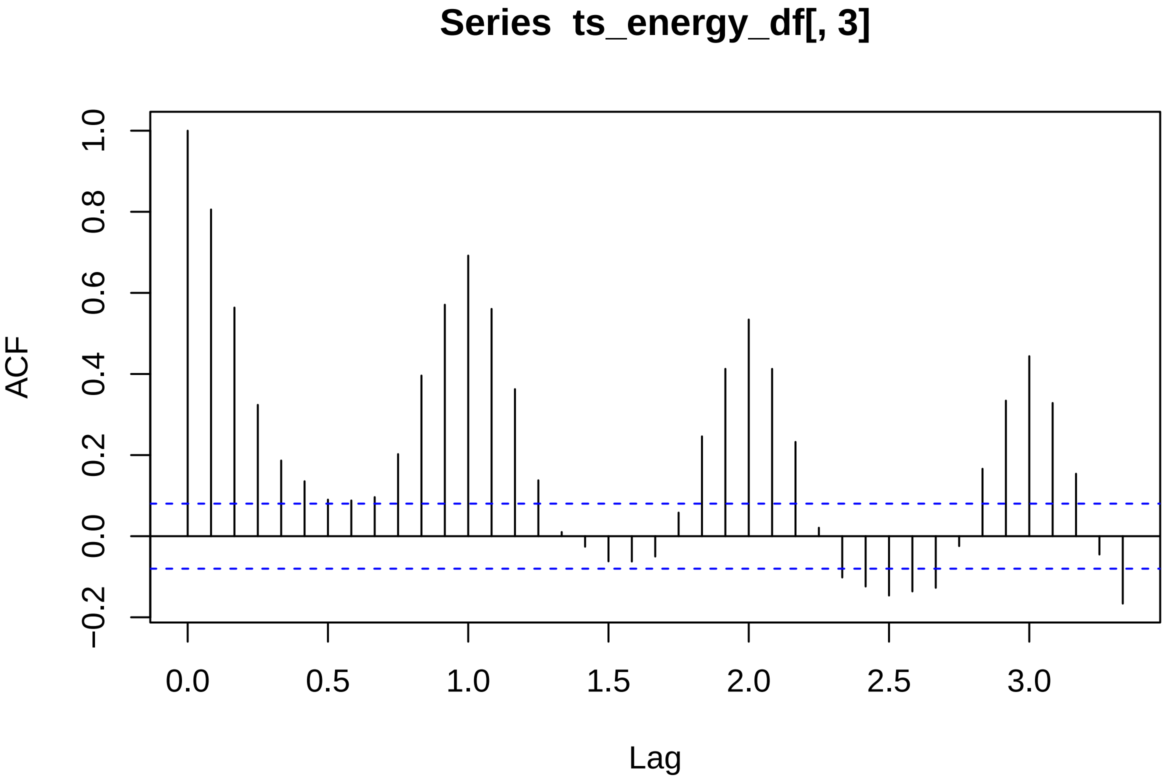
```
acf(ts_energy_df[,1], lag.max = 40)
```



```
acf(ts_energy_df[,2], lag.max = 40)
```



```
acf(ts_energy_df[,3], lag.max = 40)
```

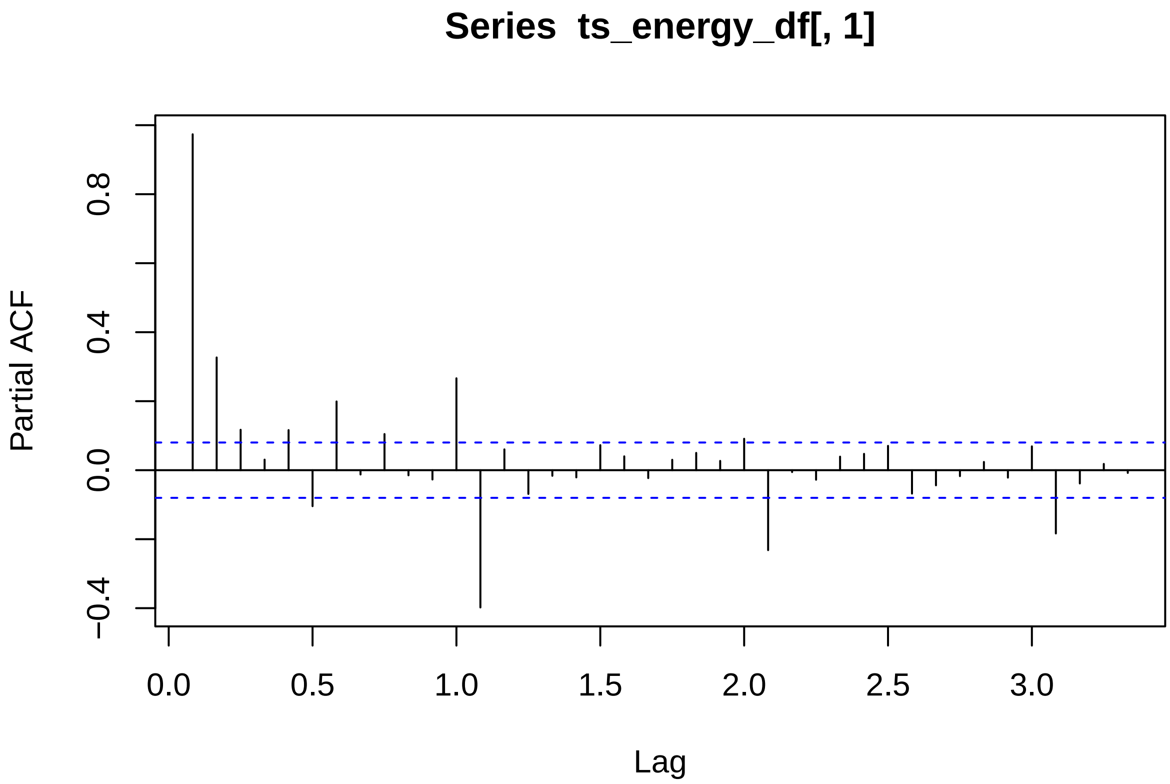


Question 7

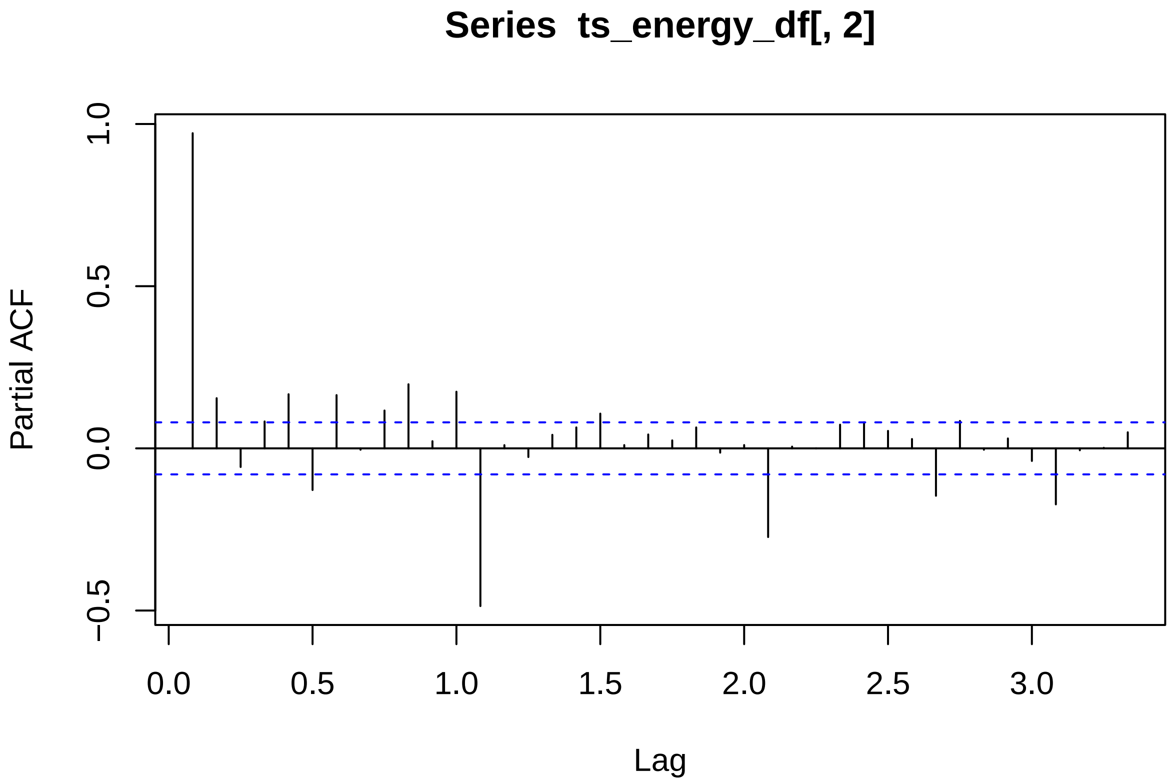
Compute the partial autocorrelation function from lag 1 to lag 40 for these three variables. How these plots differ from the ones in Q6?

These plots differ from the ones in Q6 that there are negative and positive values in the pacf graphs. Also, most values are within the blue line, showing a weak correlation and may not be significant. The partial autocorrelation sharply decreases after lag 0. The first two graphs show similar behavior, but the third graph have a spike around 0.7. All three graphs have a negative spike around lag 1.1.

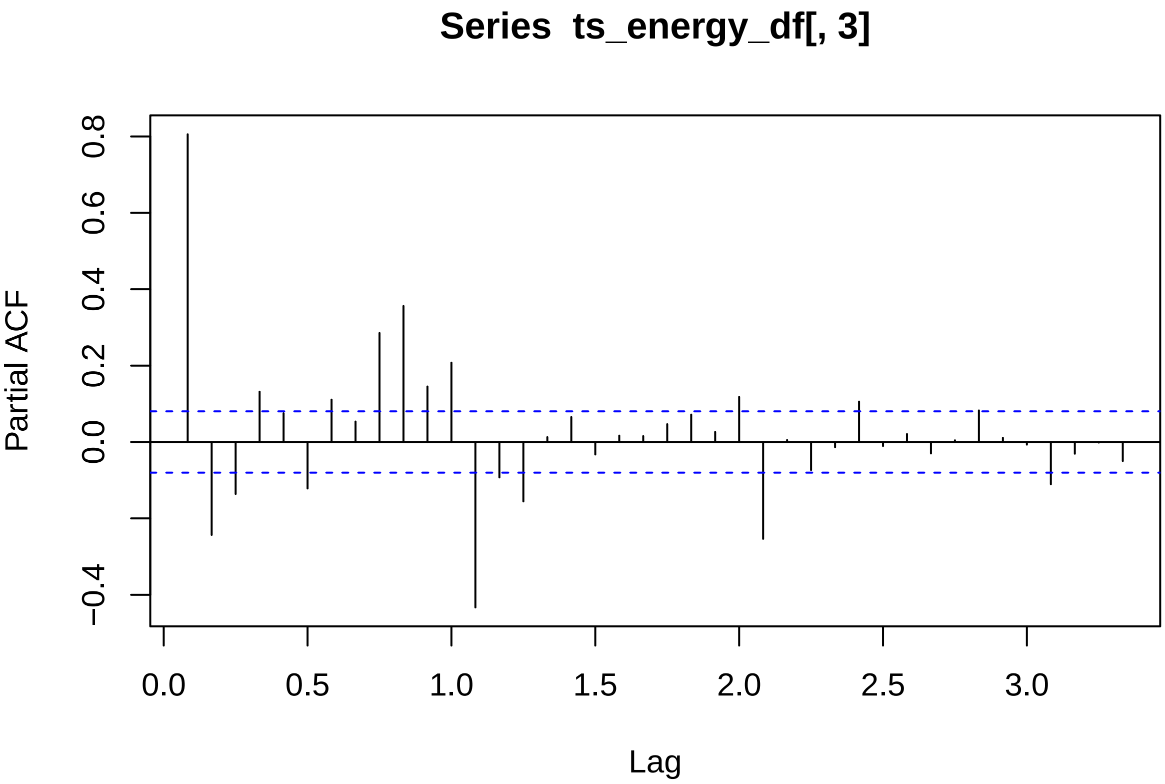
```
pacf(ts_energy_df[,1], lag.max = 40)
```



```
pacf(ts_energy_df[,2], lag.max = 40)
```



```
pacf(ts_energy_df[,3], lag.max = 40)
```


Series ts_energy_df[, 3]
Partial ACF
0.8
0.6
0.4
0.2
0.0
−0.4
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Lag