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

Assignment 2 - Due date 02/25/24

Jaimie Wargo

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_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).

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.\

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
## here() starts at C:/Users/jaimi/OneDrive/Documents/Duke/Spring_2024/TSA_Sp24
##
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

Data set information

Consider the data provided in the spreadsheet "Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source on our **Data** folder. The data comes from the US Energy Information and Administration and corresponds to the December 2023 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
raw_data <- read_excel(path=here('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‘
```

```
#Now let's extract the column names from row 11
read_col_names <- read_excel(path=here('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‘
```

```
colnames(raw_data) <- read_col_names
head(raw_data)
```

```
## # 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>, ...
```

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.

```
energy_df <- raw_data[,c('Month', 'Total Biomass Energy Consumption',
                         'Total Renewable Energy Consumption',
                         'Hydroelectric Power Consumption')]
head(energy_df)
```

```
## # A tibble: 6 x 4
##   Month                 ‘Total Biomass Energy Consumption‘ Total Renewable Energ~1
##   <dttm>                                             <dbl>                   <dbl>
## 1 1973-01-01 00:00:00                                 130.                    220.
## 2 1973-02-01 00:00:00                                 117.                    197.
## 3 1973-03-01 00:00:00                                 130.                    219.
## 4 1973-04-01 00:00:00                                 126.                    209.
## 5 1973-05-01 00:00:00                                 130.                    216.
## 6 1973-06-01 00:00:00                                 126.                    208.
## # i abbreviated name: 1: ‘Total Renewable Energy Consumption‘
## # i 1 more variable: ‘Hydroelectric Power Consumption‘ <dbl>
```

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().

```
energy_ts <- ts(energy_df[,2:4], start=c(1973, 1), frequency = 12)
head(energy_ts)
```

```
##          Total Biomass Energy Consumption Total Renewable Energy Consumption
## Jan 1973                          129.787                            219.839
## Feb 1973                          117.338                            197.330
## Mar 1973                          129.938                            218.686
## Apr 1973                          125.636                            209.330
## May 1973                          129.834                            215.982
## Jun 1973                          125.611                            208.249
##          Hydroelectric Power Consumption
## Jan 1973                          89.562
## Feb 1973                          79.544
## Mar 1973                          88.284
## Apr 1973                          83.152
## May 1973                          85.643
## Jun 1973                          82.060
```

Question 3

Compute mean and standard deviation for these three series.

```
biomass_mean <- mean(energy_ts[,1])
paste("Biomass - mean: ", round(biomass_mean,2),
      " std. dev: ", round(sd(energy_ts[,1]),2))
```

```
## [1] "Biomass - mean:  277.54  std. dev:  89.15"
```

```
totalrenew_mean <- mean(energy_ts[,2])
paste("Total Renewable Energy - mean: ", round(totalrenew_mean,2),
      " std. dev: ", round(sd(energy_ts[,2]),2))
```

```
## [1] "Total Renewable Energy - mean:  393.46  std. dev:  133.72"
```

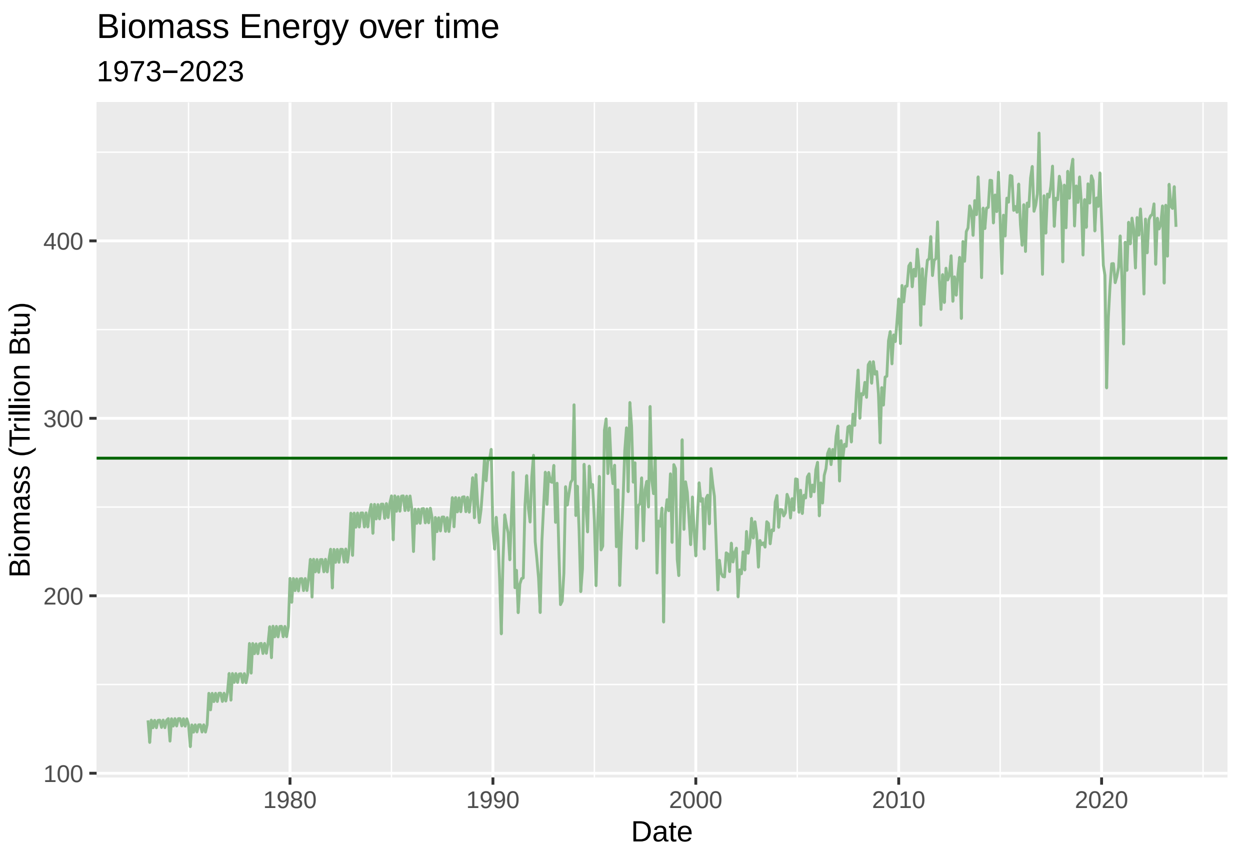
```
hydro_mean <- mean(energy_ts[,3])
paste("Hydroelectric - mean: ", round(hydro_mean,2),
      " std. dev: ", round(sd(energy_ts[,3]),2))
```

```
## [1] "Hydroelectric - mean:  79.73  std. dev:  14.15"
```

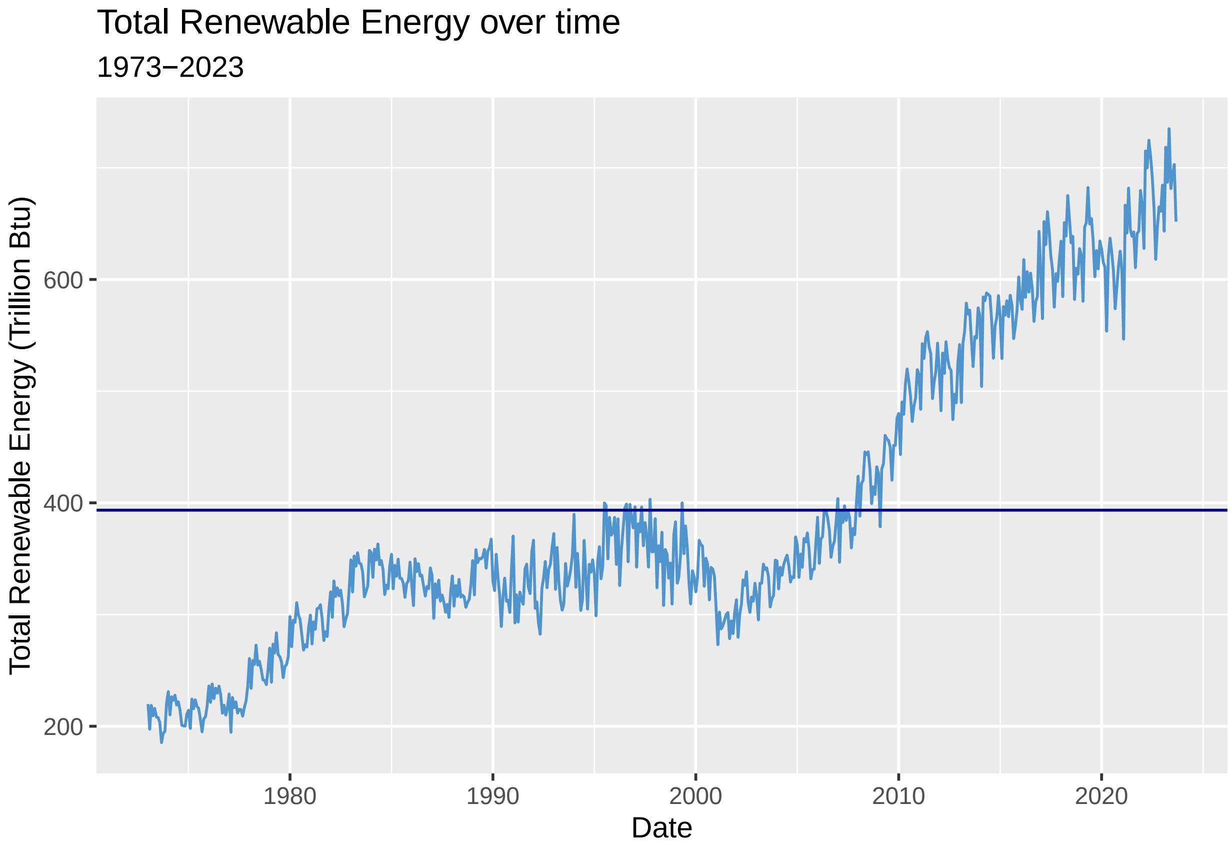
Question 4

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.

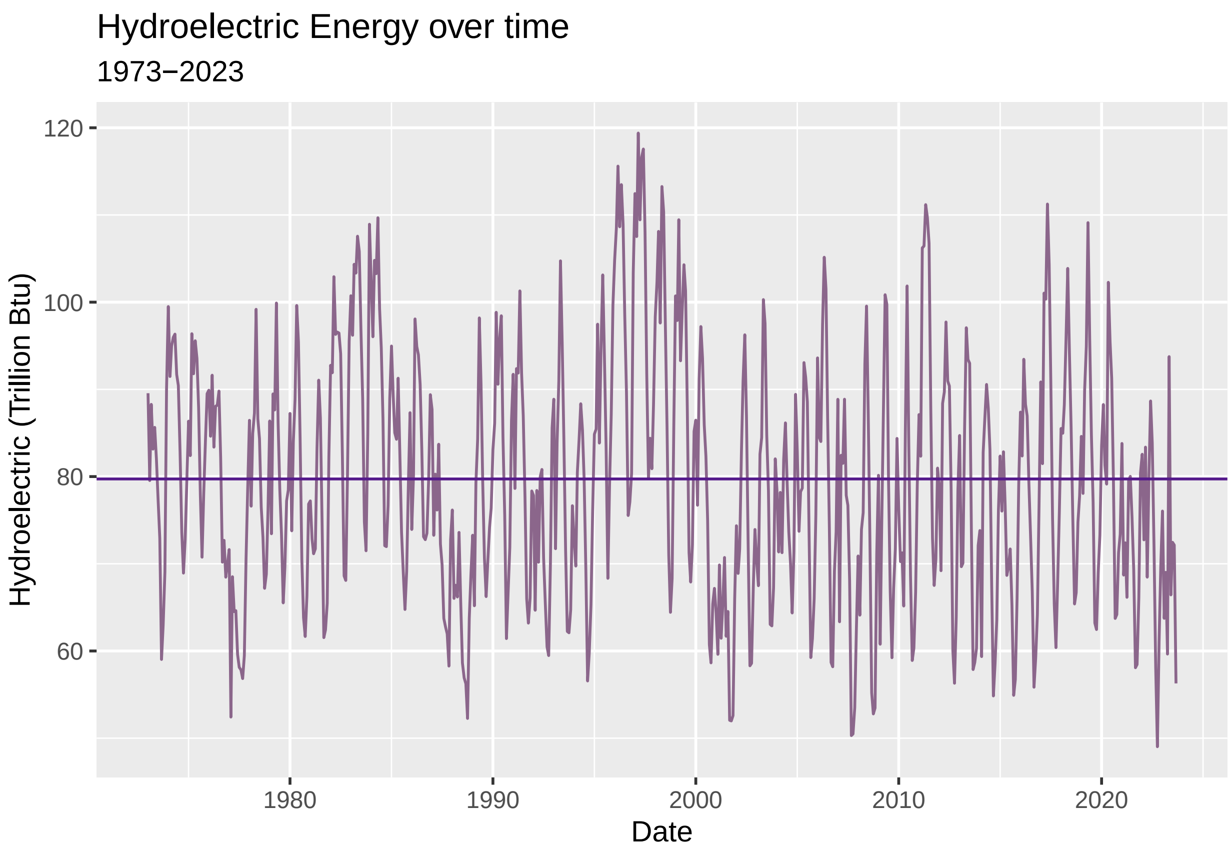
```
nospace_colnames <- c('date', 'biomass', 'total_renew', 'hydro')
colnames(energy_df) <- nospace_colnames

ggplot(data=energy_df, aes(x=date, y=biomass))+
  geom_line(color='darkseagreen')+
  geom_hline(yintercept = biomass_mean, color='darkgreen')+
  labs(x='Date', y='Biomass (Trillion Btu)',
       title='Biomass Energy over time', subtitle = '1973-2023')
```



```
ggplot(data=energy_df, aes(x=date, y=total_renew))+
  geom_line(color='steelblue3')+
  geom_hline(yintercept = totalrenew_mean, color='navy')+
  labs(x='Date', y='Total Renewable Energy (Trillion Btu)',
       title='Total Renewable Energy over time', subtitle = '1973-2023')
```



```r
ggplot(data=energy_df, aes(x=date, y=hydro))+
  geom_line(color='plum4')+
  geom_hline(yintercept = hydro_mean, color='purple4')+
  labs(x='Date', y='Hydroelectric (Trillion Btu)',
       title='Hydroelectric Energy over time', subtitle = '1973-2023')
```



Question 5

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

```r
paste("Biomass and Total Renewable:", round(cor(energy_ts[,1], energy_ts[,2]),3))
```

```
## [1] "Biomass and Total Renewable: 0.967"
```

```r
paste("Biomass and Hydro:", round(cor(energy_ts[,2], energy_ts[,3]),3))
```

```
## [1] "Biomass and Hydro: 0"
```

```r
paste("Hydro and Total Renewable:", round(cor(energy_ts[,1], energy_ts[,3]),3))
```

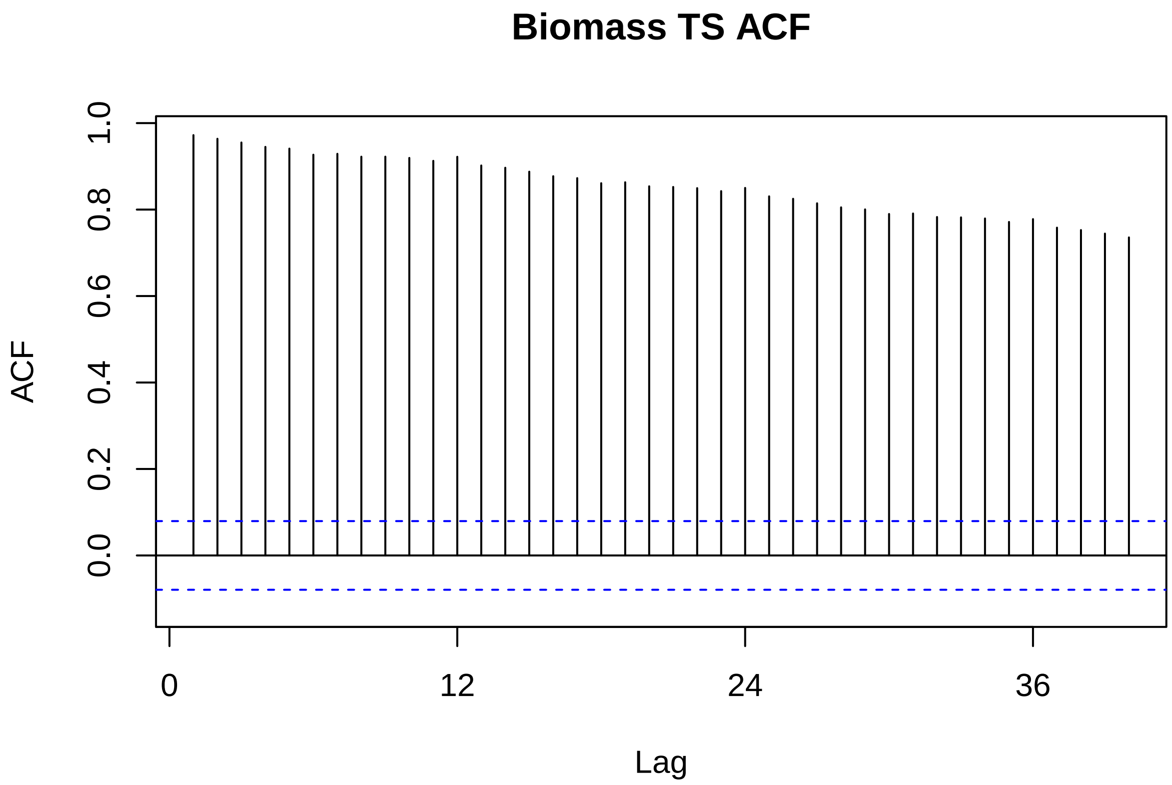
```
## [1] "Hydro and Total Renewable: -0.098"
```

Answer: The biomass and total renewable energy series are strongly correlated, with an R2 value of 0.967. Hydroelectric power is not strongly correlated to biomass or renewable energy, with R2 values of 0 and -0.098 respectively. This makes sense because hydroelectric power consumption is more dependent on seasonality and resource availability and is less representative of the general increase in renewable energy over time.

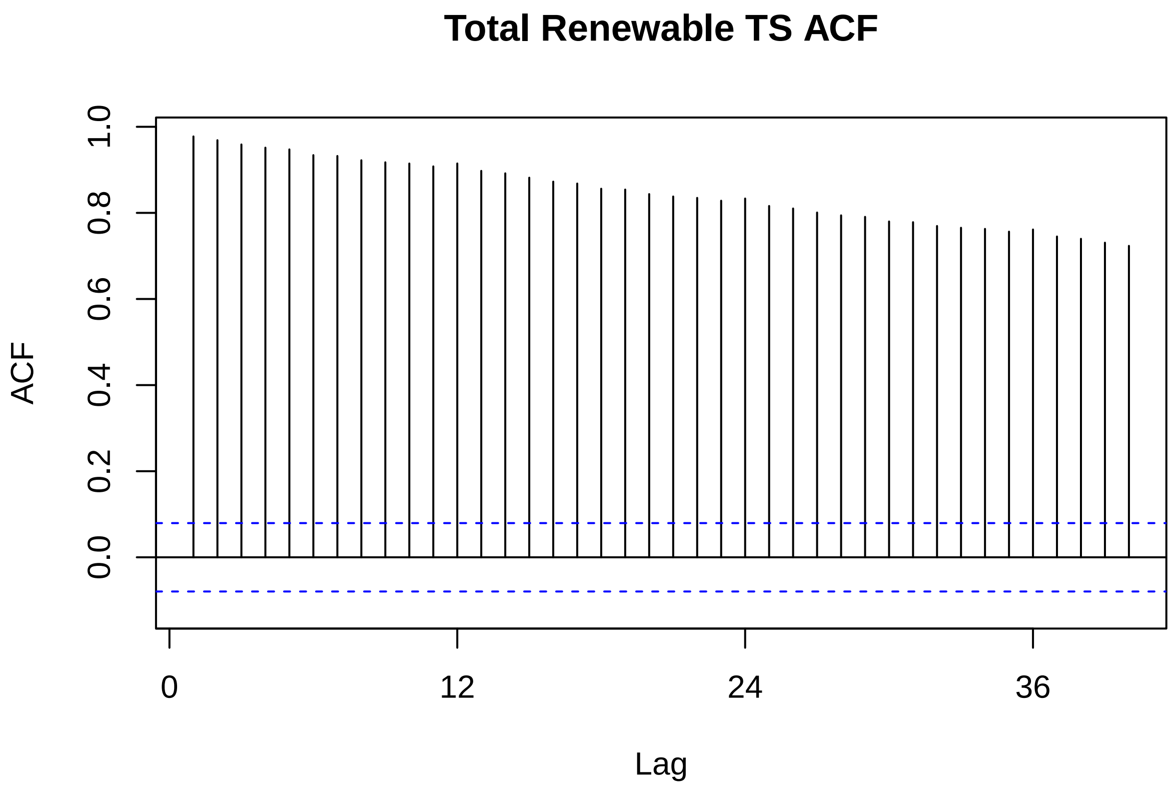
Question 6

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?

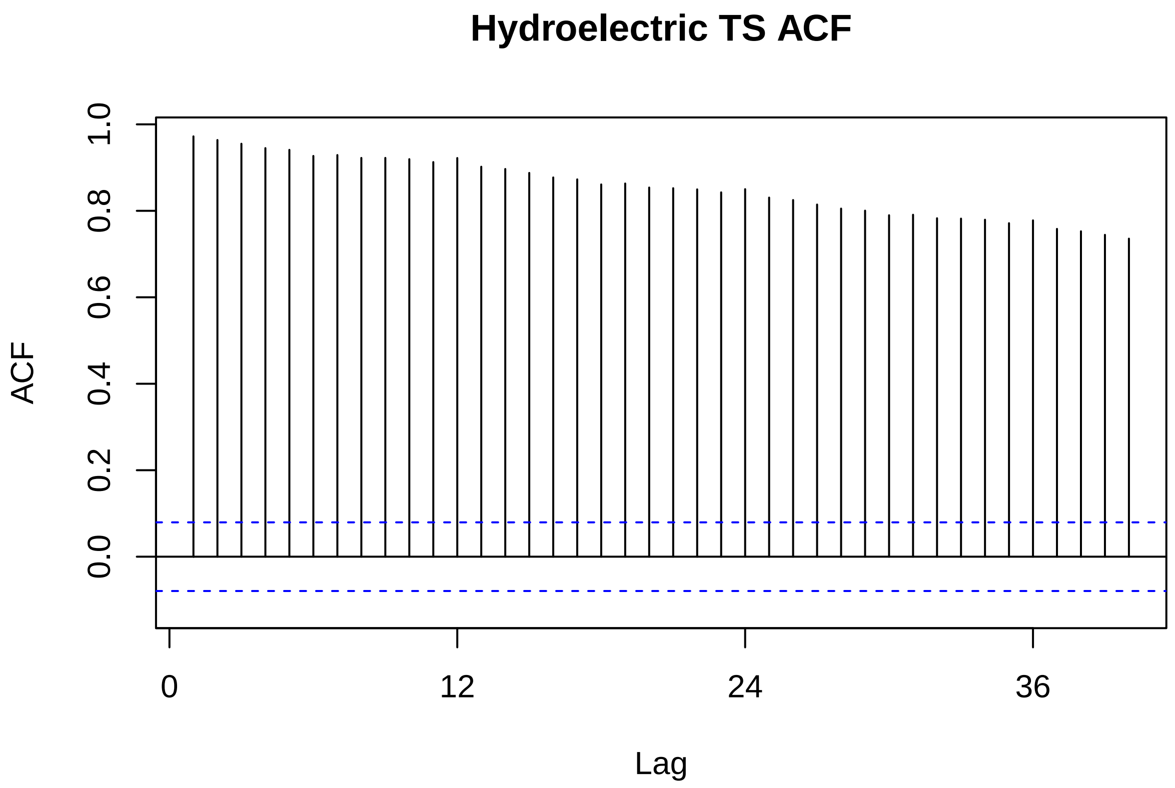
```
Acf(energy_ts[,1], lag.max=40, main="Biomass TS ACF")
```



```
Acf(energy_ts[,2], lag.max=40, main="Total Renewable TS ACF")
```



```
Acf(energy_ts[,1], lag.max=40, main="Hydroelectric TS ACF")
```

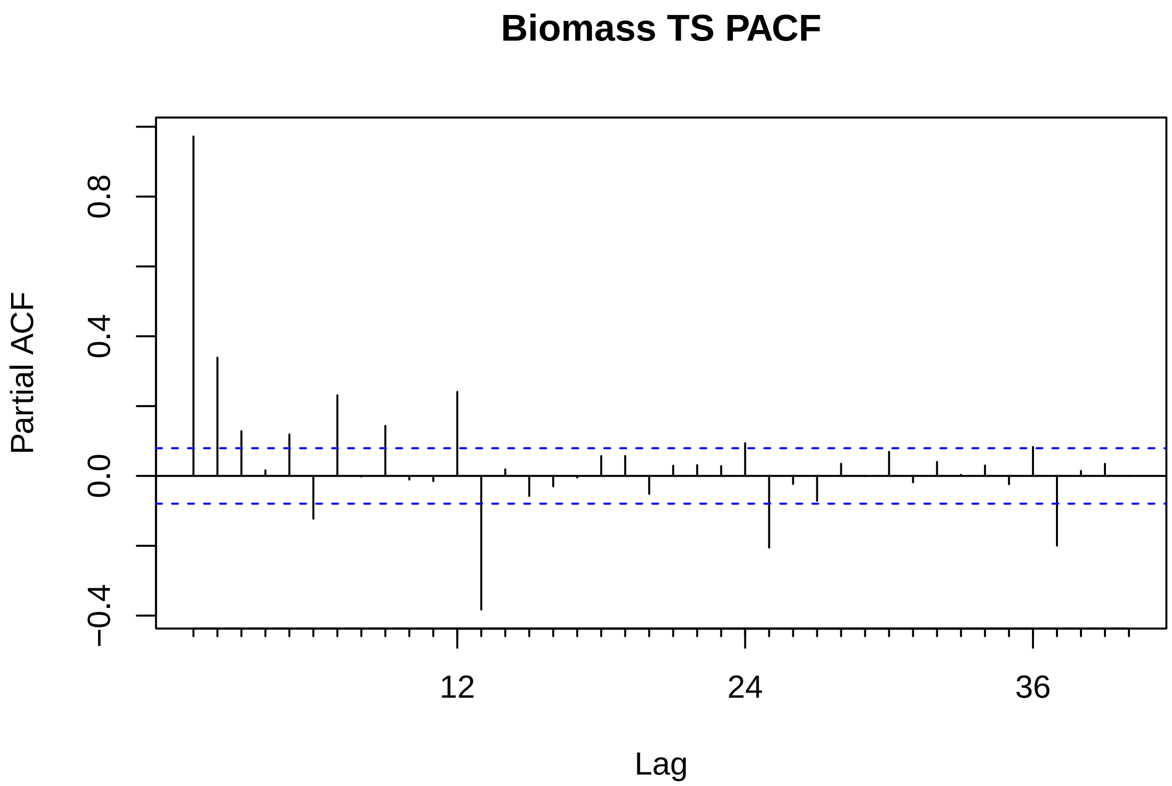


They do have the same behavior, as all of these plots show high autocorrelation through lag 40, although there is a slight decrease in ACF over time. This indicates that these observations are dependent on the intermediate observations, and a PACF plot would be helpful to eliminate this dependency.

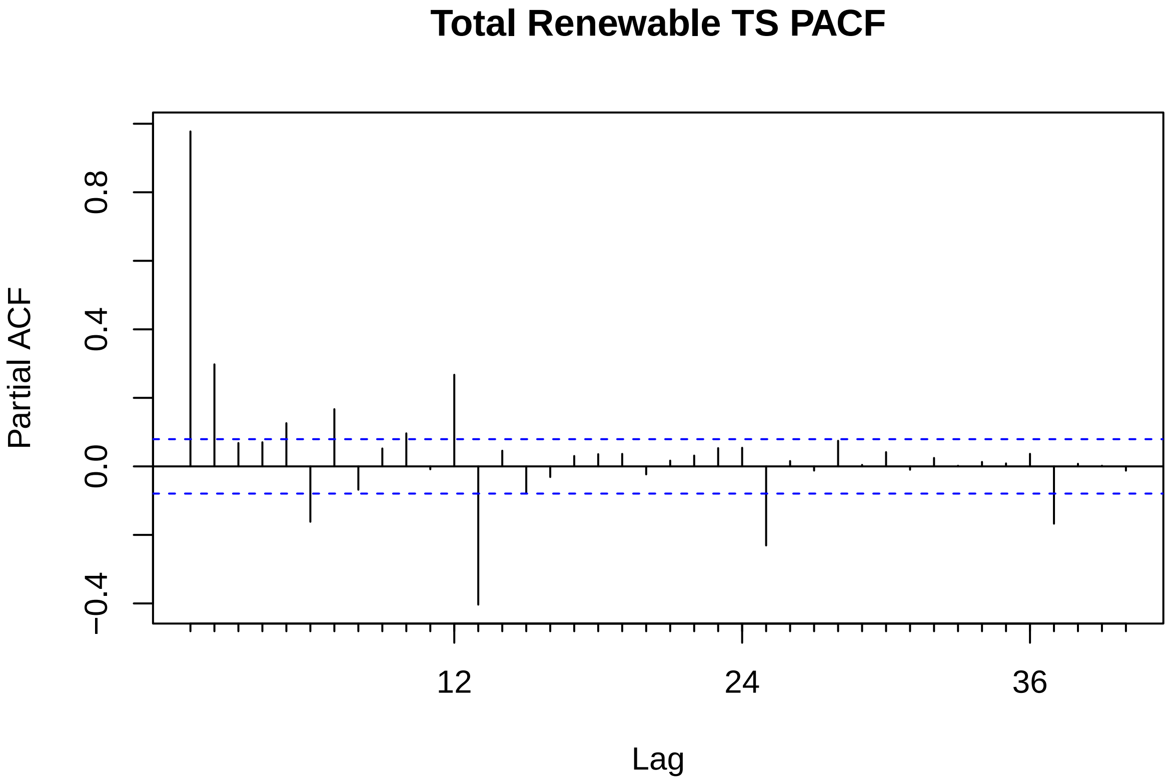
Question 7

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

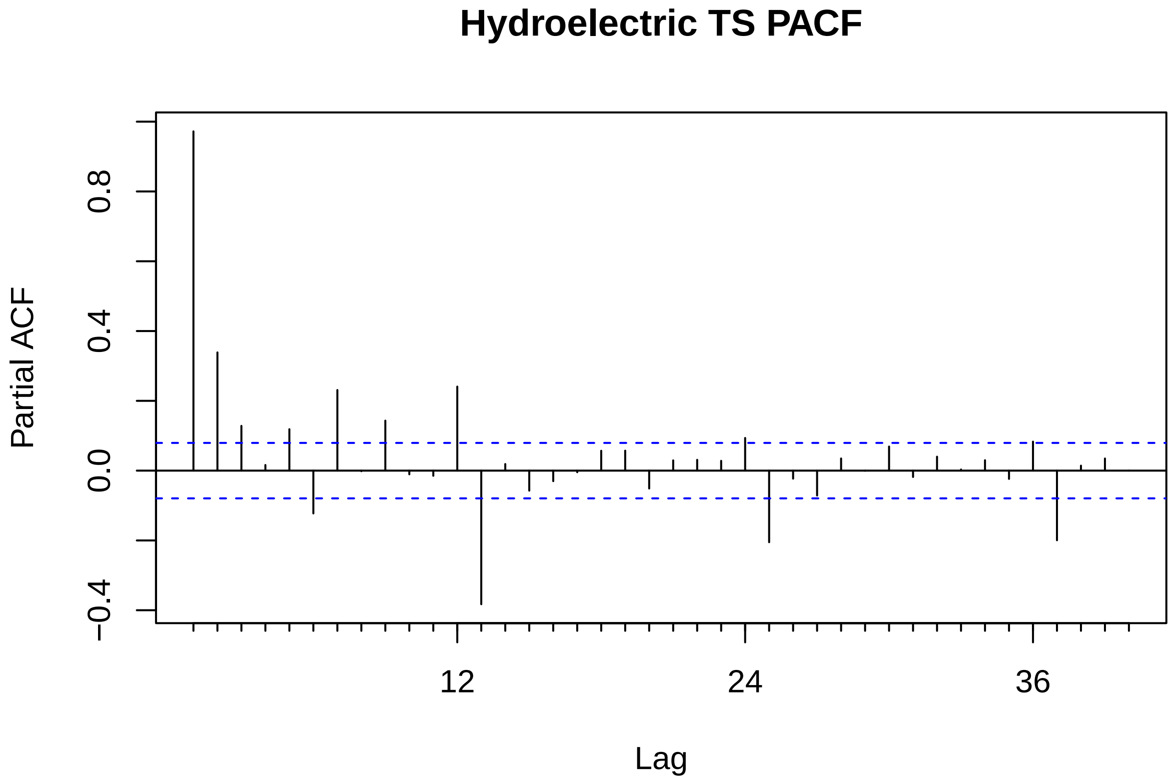
```
Pacf(energy_ts[,1], lag.max=40, main="Biomass TS PACF")
```



```
Pacf(energy_ts[,2], lag.max=40, main="Total Renewable TS PACF")
```



```r
Pacf(energy_ts[,1], lag.max=40, main="Hydroelectric TS PACF")
```



These plots are much different because they remove the intermediate variable dependency, so now there are high PACF values only at lag 1, with points for consideration at lag 2, 12, and 13. This is different from the plots in Q6 because they do not all remain high.