ENV 790.30 - Time Series Analysis for Energy Data | Spring 2021

Assignment 2 - Due date 01/26/22

Ben Joseph

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 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. Rename the pdf file such that it includes your first and last name (e.g., "LuanaLima_TSA_A02_Sp22.Rmd"). 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
#install.packages("forecast")
#install.packages("tseries")
#install.packages("dplyr")
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

```
library(forecast)
```

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

```
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
```

```r
library(tseries)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v readr   2.1.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Data set information

Consider the data provided in the spreadsheet "Table_10.1_Renewable_Energy_Production_and_ Consumption_by_Source.xlsx" on our **Data** folder. The data comes from the US Energy Information and Administration and corresponds to the January 2022 Monthly Energy Review. The spreadsheet is ready to be used. Use the command *read.table()* to import the data in R or *panda.read_excel()* in Python (note that you will need to import pandas package). }

```r
library(readxl)
df <- read_excel("./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx"
   , sheet = 1, col_names = TRUE, skip =10, na="Not Available")
```

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.

```r
df2 <- df[2:586,4:6]
head(df2,10)
```

```
## # A tibble: 10 x 3
##    `Total Biomass Energy Production` `Total Renewable Ener~ `Hydroelectric Powe~
##    <chr>                             <chr>                  <chr>
##  1 129.787                           403.981                272.703
##  2 117.338                           360.9                  242.199
##  3 129.938                           400.161                268.81
##  4 125.636                           380.47                 253.185
##  5 129.834                           392.141                260.77
##  6 125.611                           377.232                249.859
##  7 129.787                           367.325                235.67
##  8 129.918                           353.757                222.077
##  9 125.782                           307.006                179.733
## 10 129.97                            323.453                191.723
```

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().

```
# I had to convert data to numeric to transform it to a time series object
df2$`Total Biomass Energy Production`<- as.numeric(df2$`Total Biomass Energy Production`)
df2$`Total Renewable Energy Production`<- as.numeric(df2$`Total Renewable Energy Production`)
df2$`Hydroelectric Power Consumption`<-as.numeric(df2$`Hydroelectric Power Consumption`)
ts <- ts(data = df2, start = 1973, frequency = 12)
head(ts,10)
```

```
##          Total Biomass Energy Production Total Renewable Energy Production
## Jan 1973                         129.787                           403.981
## Feb 1973                         117.338                           360.900
## Mar 1973                         129.938                           400.161
## Apr 1973                         125.636                           380.470
## May 1973                         129.834                           392.141
## Jun 1973                         125.611                           377.232
## Jul 1973                         129.787                           367.325
## Aug 1973                         129.918                           353.757
## Sep 1973                         125.782                           307.006
## Oct 1973                         129.970                           323.453
##          Hydroelectric Power Consumption
## Jan 1973                         272.703
## Feb 1973                         242.199
## Mar 1973                         268.810
## Apr 1973                         253.185
## May 1973                         260.770
## Jun 1973                         249.859
## Jul 1973                         235.670
## Aug 1973                         222.077
## Sep 1973                         179.733
## Oct 1973                         191.723
```

Question 3

Compute mean and standard deviation for these three series.

```
# I had originally written out print functions that present the mean and sd for
# each function but the code and outputs were so long they ran off the page so I
# found code that creates a table for mean and sd and commented out the original code

#print(paste0("The mean of annual biomass energy production is ",round(mean(df2$`Total Biomass Energy P
#print(paste0("The mean of annual reneweable energy production is ", round(mean(df2$`Total Renewable En
#print(paste0("The mean of annual hydroelectric power consumption is ", round(mean(df2$`Hydroelectric P

#code source https://stackoverflow.com/questions/20794284/means-and-sd-for-columns-in-a-dataframe-with-
as.data.frame( t(sapply(ts, function(cl) list(Means=mean(cl,na.rm=TRUE),
    StandardDeviation=sd(cl,na.rm=TRUE)))))
```

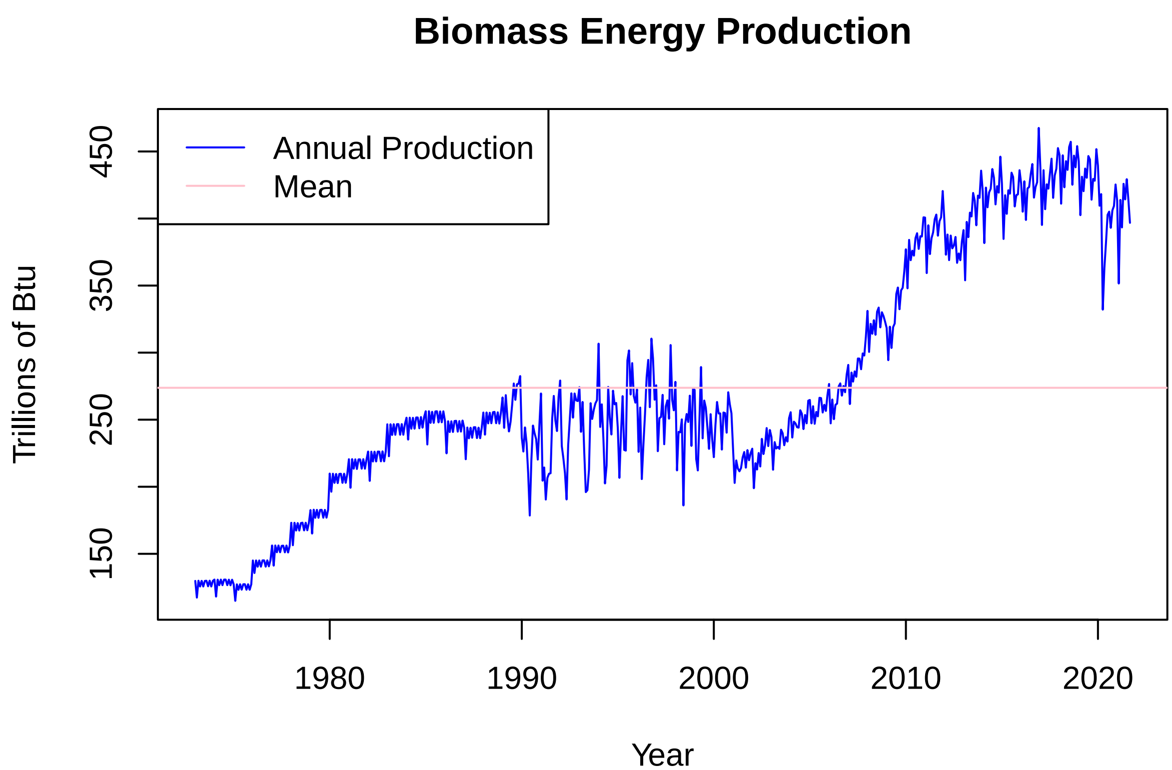
```
##                                    Means StandardDeviation
```

```
## Total Biomass Energy Production   273.7839          89.42852
## Total Renewable Energy Production 581.1708          177.5607
## Hydroelectric Power Consumption   235.9653          44.01749
```

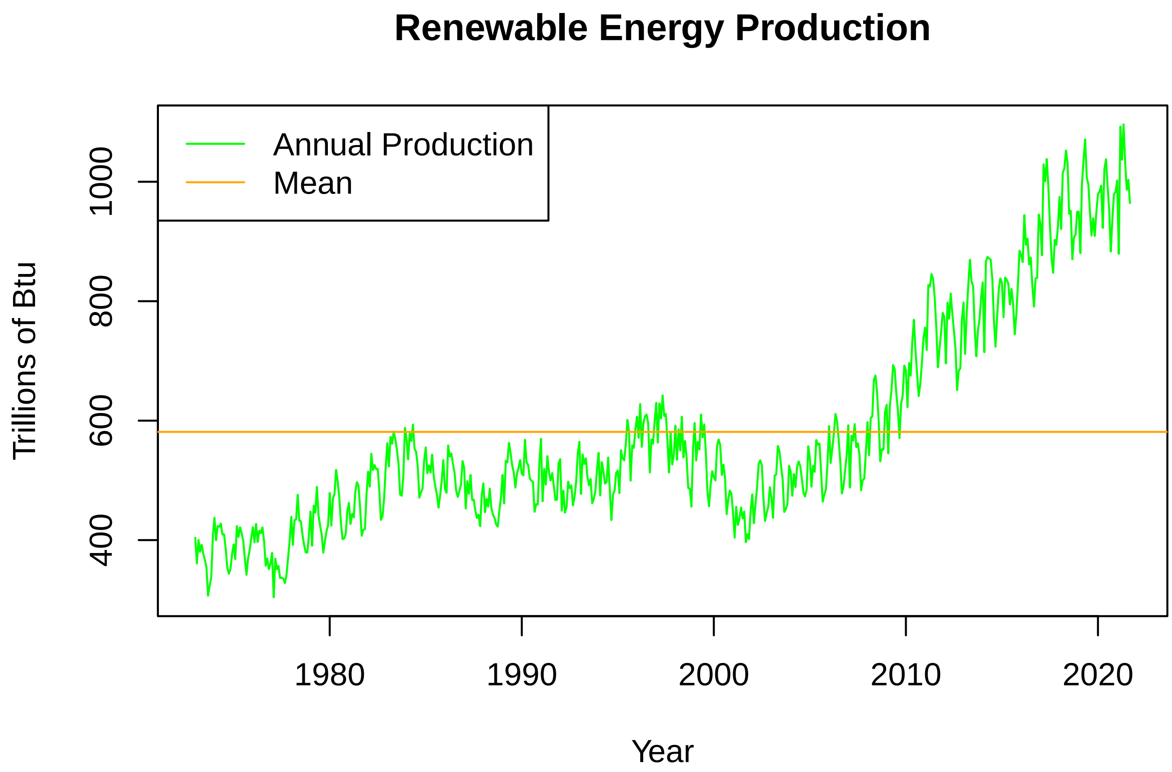
Question 4

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.

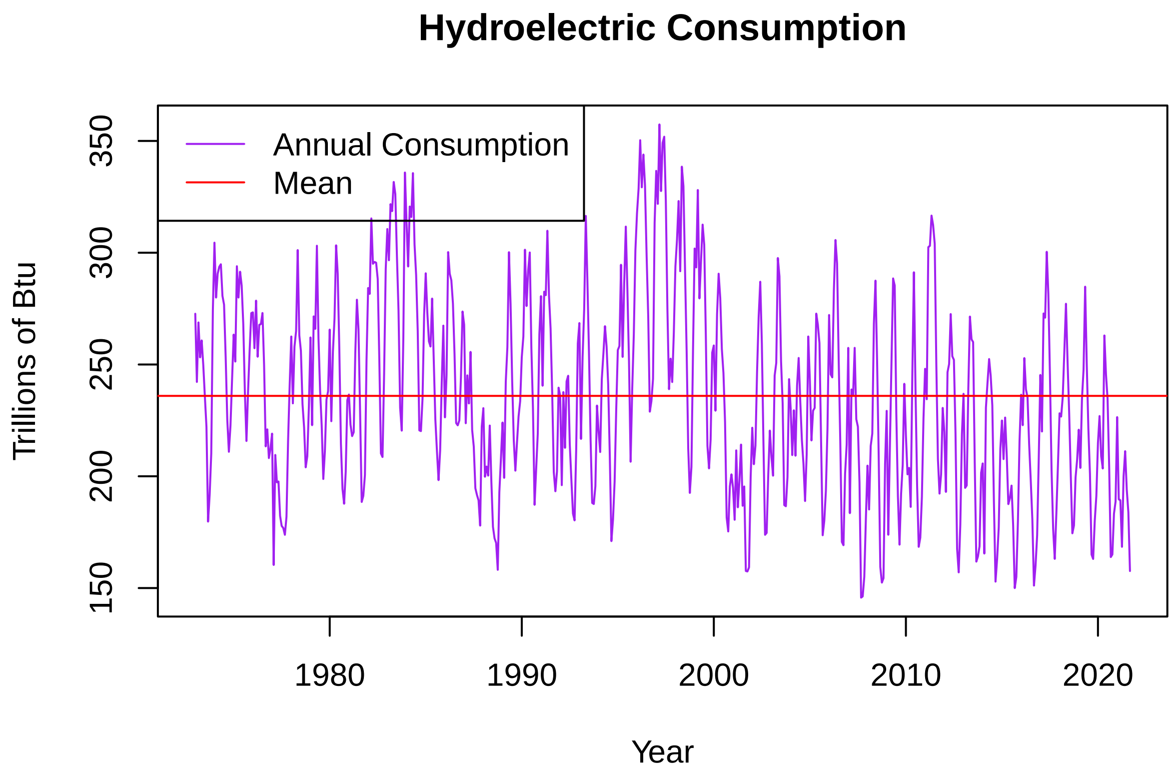
```
#Biomass Plot
plot(ts[,1], col="blue", ylab="Trillions of Btu", main="Biomass Energy Production",
     xlab="Year")
abline(h=mean(ts[,1]),col="pink")
legend("topleft",legend=c("Annual Production","Mean"), lty=c("solid","solid"),
       col = c("blue","pink"))
```



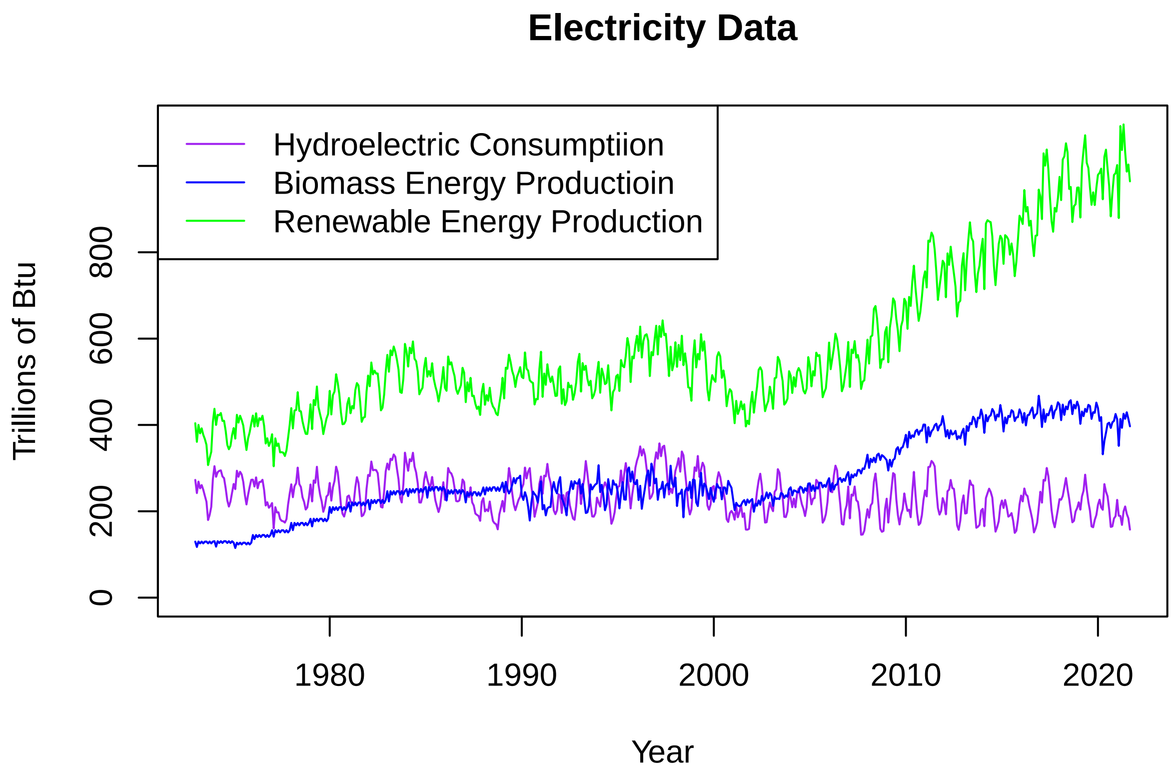

```
#Renewable Plot
plot(ts[,2], col="green", ylab="Trillions of Btu", main="Renewable Energy Production",
     xlab="Year")
abline(h=mean(ts[,2]),col="orange")
legend("topleft",legend=c("Annual Production","Mean"), lty=c("solid","solid"),
       col = c("green","orange"))
```




```
#Hydro Plot
plot(ts[,3], col="purple", ylab="Trillions of Btu", main="Hydroelectric Consumption",
     xlab="Year")
abline(h=mean(ts[,3]),col="red")
legend("topleft",legend=c("Annual Consumption","Mean"), lty=c("solid","solid"),
       col = c("purple","red"))
```




```r
#Adding biomass and renewable energy lines to hydro plot, renaming it, setting y limit to
#equal max value across three series and adding legend
plot(ts[,3], col="purple", ylab="Trillions of Btu", main="Electricity Data", xlab="Year",
     ylim=c(0,(max(c(df2$`Total Biomass Energy Production`,df2$`Total Renewable Energy Production`,
    df2$`Hydroelectric Power Consumption`)))))
lines(ts[,1],col="blue")
lines(ts[,2],col="green")
legend("topleft",legend=c("Hydroelectric Consumptiion","Biomass Energy Productioin",
    "Renewable Energy Production"), lty=c("solid","solid","solid"),
    col = c("purple","blue","green"))
```



Question 5

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

I used the cor.test function to test correlation between the three series. The values of interest are the correlation coefficient ("cor) and the p-value displayed in the output. The correlation coefficient shows the strength and direction of the correlation. The closer to 1, the stronger the positive correlation. The closer to -1, the stronger the negative correlation.

The p-value shows the probability of observing a correlation coefficient from a given sample that is at least as extreme as the observed correlation coefficient if we were to assume there is no correlation between the two series. If the p value is below a significance level of 5%, or .05, we assume that the observed correlation did not occur by chance, and that there is a statistically significant correlation between the series.

There is a strong positive correlation between biomass and renewable energy production with a correlation coefficient near 1 and a very small p value. There is a weaker, but still statistically significant correlation between biomass energy production and hydroelectric power consumption. There is no statistically significant correlation between renewable enery production and hydroelectric power production as evidenced by a p-value well above .05. For exact values, see the outputs below for the three series being compared to each other, two at a time.

```r
print("Correllation analysis for Biomass Energy Production and Renewable Energy Production:")
```

```
## [1] "Correllation analysis for Biomass Energy Production and Renewable Energy Production:"
```

```r
cor.test(ts[,1],ts[,2], use = "everything", method = c("pearson","kendall","spearman"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  ts[, 1] and ts[, 2]
## t = 58.037, df = 583, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9103552 0.9344118
## sample estimates:
```

```
##       cor
## 0.9232838
```

```r
print("Correllation analysis for Biomass Energy Production and Hydroelectric Power Consumption:")
```

```
## [1] "Correllation analysis for Biomass Energy Production and Hydroelectric Power Consumption:"
```

```r
cor.test(ts[,1],ts[,3], use = "everything", method = c("pearson","kendall","spearman"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  ts[, 1] and ts[, 3]
## t = -7.056, df = 583, p-value = 4.879e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3535258 -0.2040752
## sample estimates:
##        cor
## -0.2804997
```

```r
print("Correllation analysis for Renewable Energy Production and Hydroelectric Power Consumption:")
```

```
## [1] "Correllation analysis for Renewable Energy Production and Hydroelectric Power Consumption:"
```

```r
cor.test(ts[,2],ts[,3], use = "everything", method = c("pearson","kendall","spearman"))
```

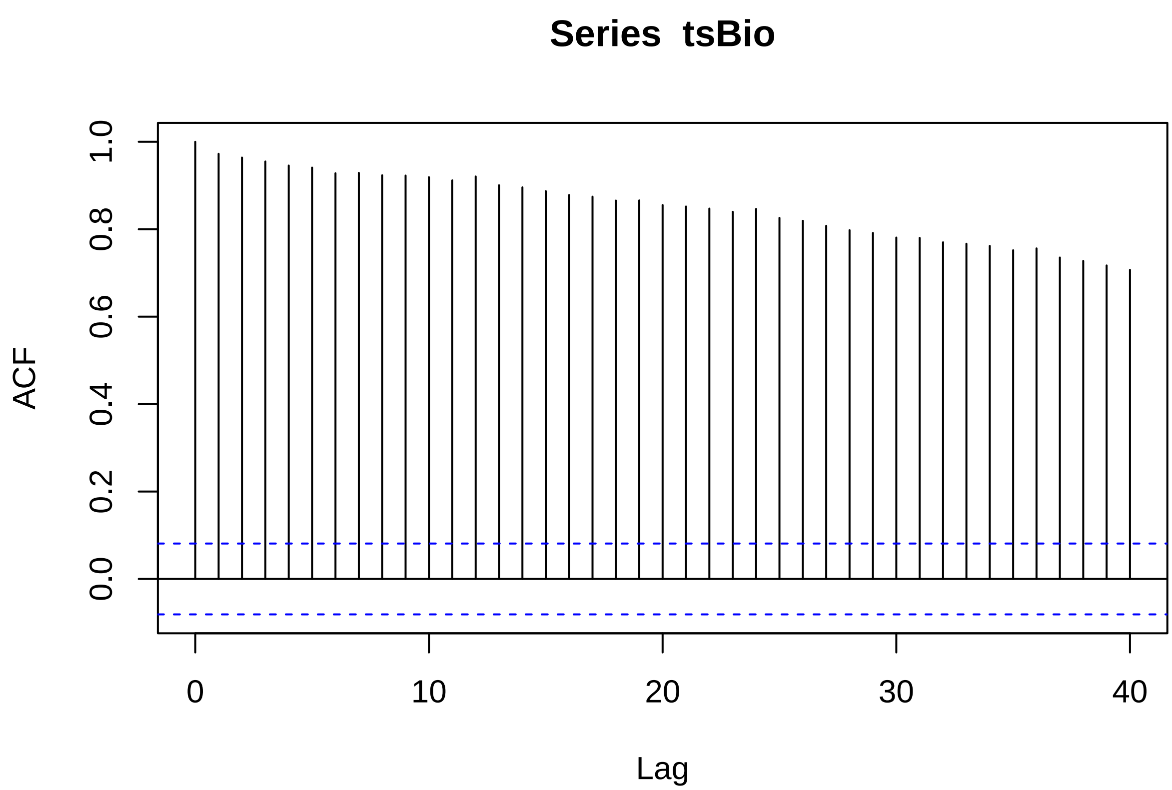
```
##
##  Pearson's product-moment correlation
##
## data:  ts[, 2] and ts[, 3]
## t = -1.3738, df = 583, p-value = 0.17
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.13723936  0.02437056
## sample estimates:
##         cor
## -0.05680651
```

Question 6

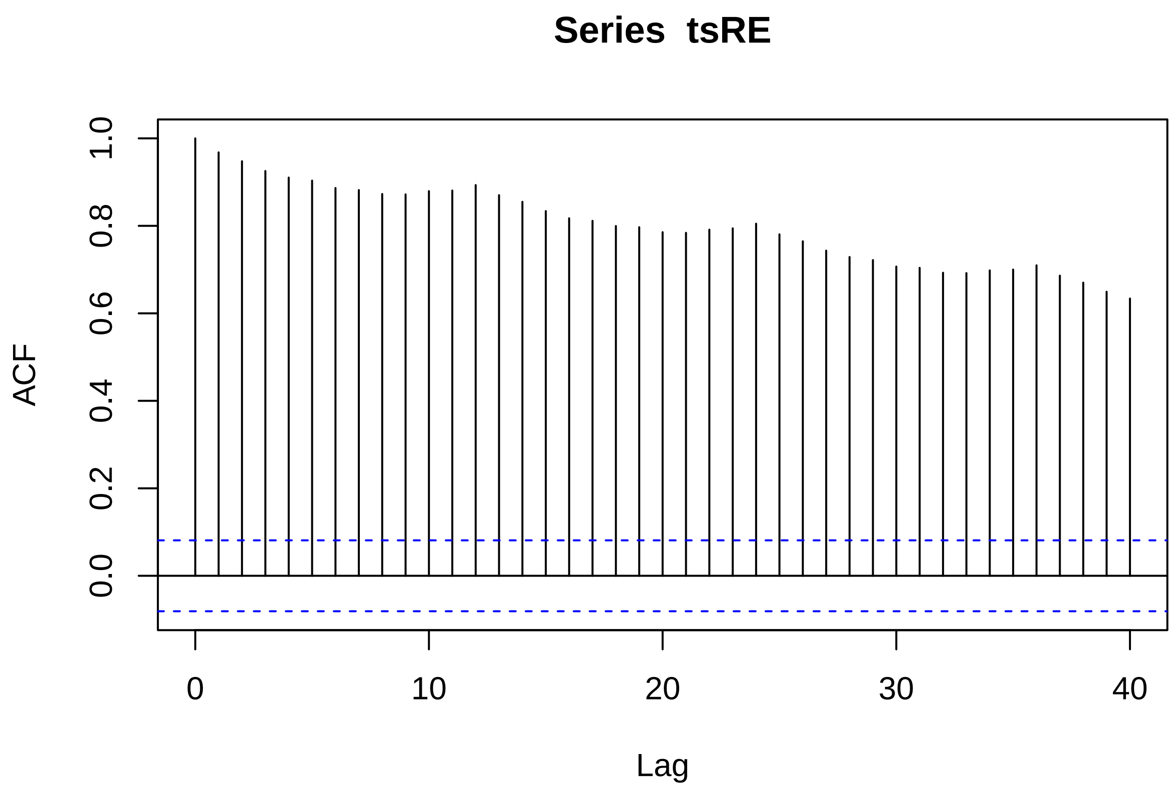
Compute the autocorrelation function from lag 1 up to lag 40 for these three variables. What can you say about these plots? Do the three of them have the same behavior?

The Biomass and Renewable energy plots have similar behavior, starting at near 1 and fairly steadily declining to around .7 by lag 40. The Renewable Energy plot shows slight seasonal trend which makes sense because renewable energy tends to be weather depenent. The hydroelectric power consumption graph shows extreme seasonal variation with an overall slight downward trend. This makes sense because the availability of hydroelectric resources depends heeavily on the abundance of water which has seasonal elements like the presence of snow runoff in the spring and early summer.

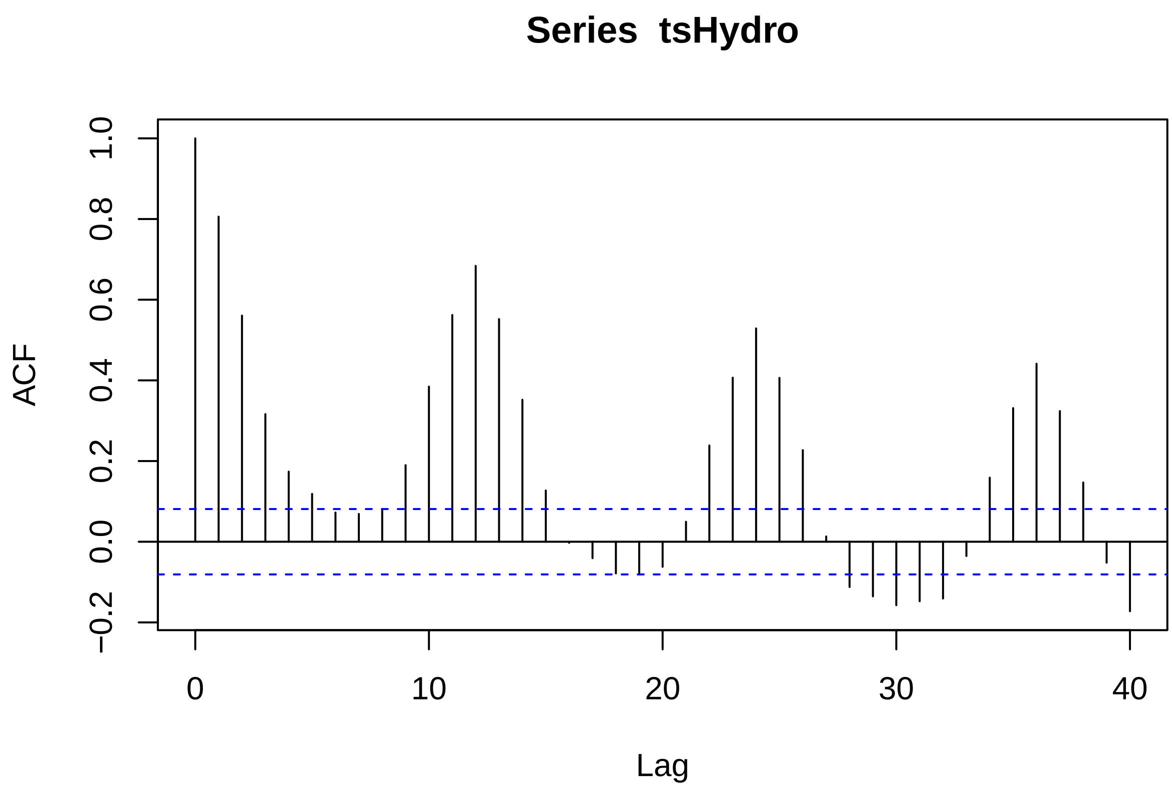
```r
# first create 3 different time series for each key variable
tsBio <- ts(data = df2$`Total Biomass Energy Production`, start = 1973, frequency = 1)
tsRE <- ts(data = df2$`Total Renewable Energy Production`, start = 1949, frequency = 1)
tsHydro <- ts(data = df2$`Hydroelectric Power Consumption`, start = 1949, frequency = 1)
acfBio <- acf(tsBio, lag.max = 40, type = "correlation",plot=TRUE)
```




```r
acfRE <- acf(tsRE, lag.max = 40, type = "correlation", plot=TRUE)
```




```
acfHydro <- acf(tsHydro, lag.max = 40, type = "correlation", plot=TRUE)
```

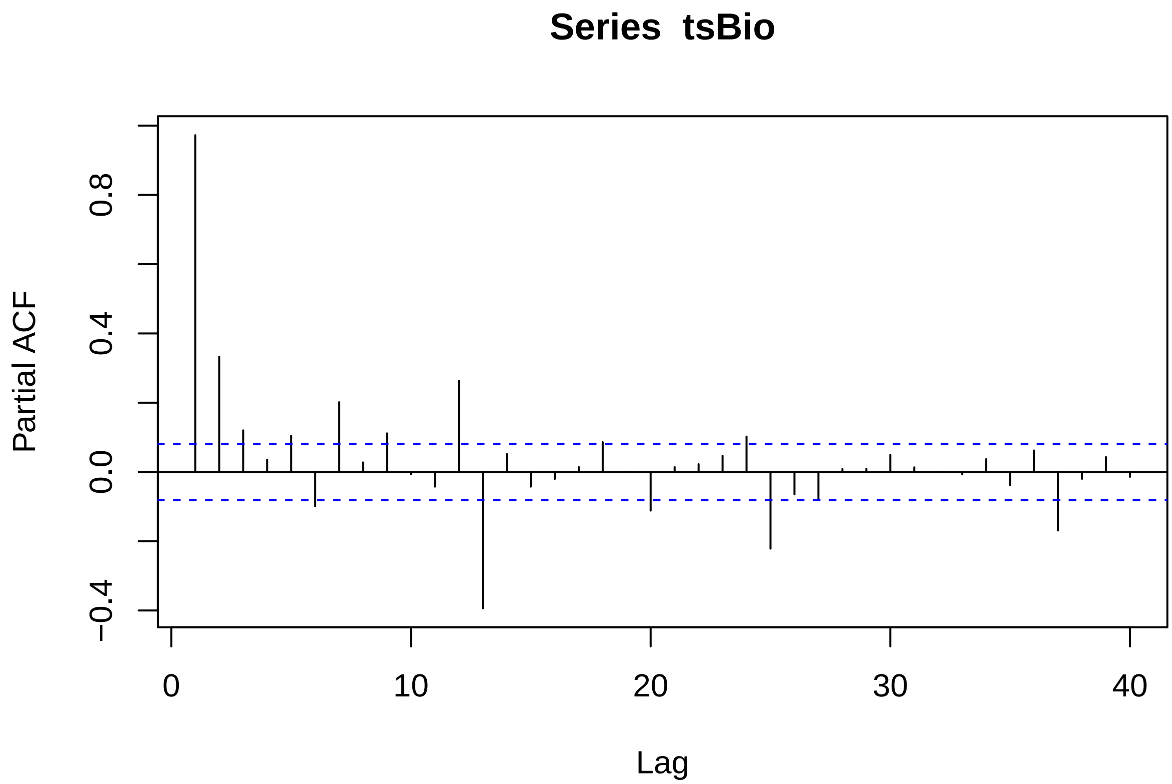


Question 7

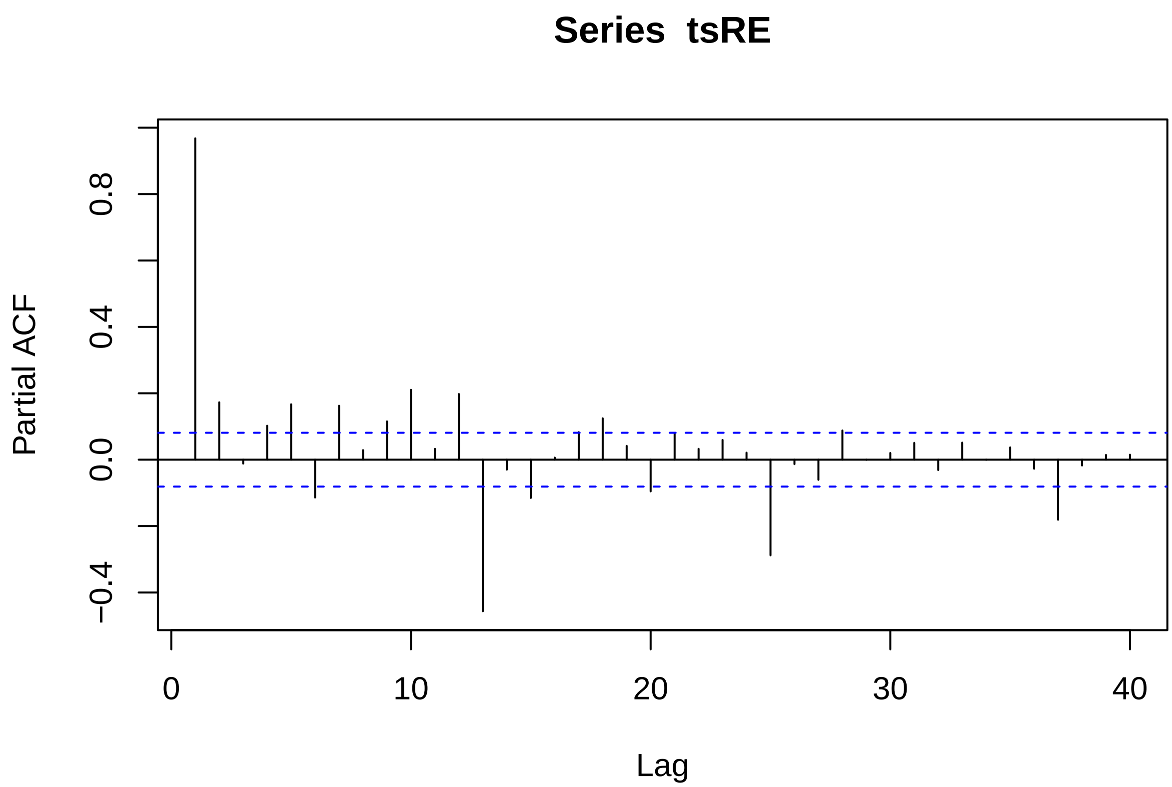
Compute the partial autocorrelation function from lag 1 to lag 40 for these three variables. How these plots differ from the ones in Q6?

As expected, the partial autocorrelation values are the same at lag 1 as they were for the autocorrelation charts because there are no intermediary values. After, lag 1, PACF values hover around 0 for all three. The only times that the partial autocorreelation values cross the blue dotted lines that show significance levels is around lag 12 and 24 which again reflects the seasonality of these resources.

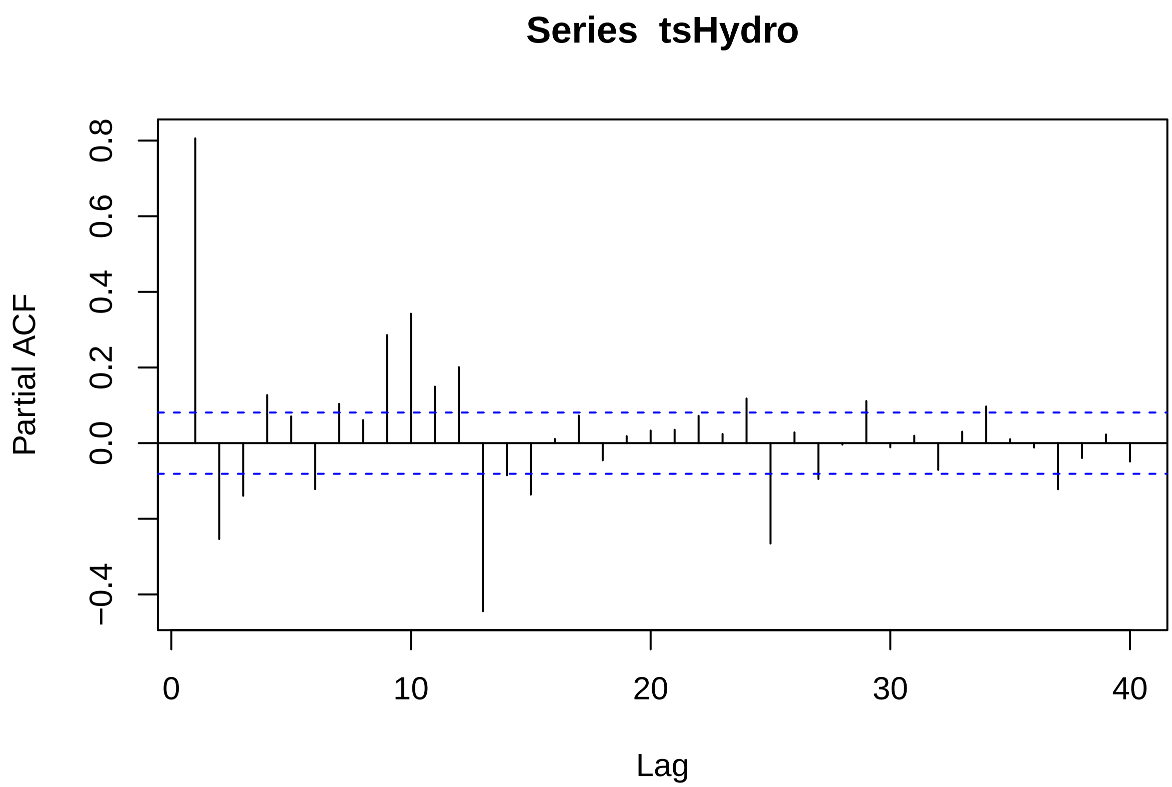
```r
pacf(tsBio, lag.max = 40, plot = T)
```




```r
pacf(tsRE, lag.max = 40, plot = T)
```




```r
pacf(tsHydro, lag.max = 40, plot = T)
```



Series tsHydro
Partial ACF
0.8
0.6
0.4
0.2
0.0
−0.4
0
10
20
30
40
Lag