ENV 790.30 - Time Series Analysis for Energy Data | Spring 2022

Assignment 4 - Due date 02/17/22

Ben Joseph

Directions

You should open the .rmd file corresponding to this assignment on RStudio. The file is available on our class repository on Github. And to do so you will need to fork our repository and link it to your RStudio.

Once you have the project open the first thing you will do is change "Student Name" on line 3 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_A04_Sp21.Rmd"). Submit this pdf using Sakai.

R packages needed for this assignment: "xlsx" or "readxl", "ggplot2", "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("xlsx")
#library(xlsx)
library(readxl)
library(ggplot2)
library(tseries)
```

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

```
library(Kendall)
```

Questions

Consider the same data you used for A3 from the spreadsheet "Table_10.1_Renewable_Energy_Production_and_Consumpti
The data comes from the US Energy Information and Administration and corresponds to the January 2021 Monthly Energy Review. For this assignment you will work only with the column "Total Renewable Energy Production".

```
#Importing data set - using xlsx package
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")
my_date <- df[2:586,1]
df <- df[2:586,5]
```

```
df <- cbind(my_date, df)
df$`Total Renewable Energy Production`<- as.numeric(df$`Total Renewable Energy Production`)
head(df,10)
```

```
##         Month Total Renewable Energy Production
## 1  1973-01-01                            403.981
## 2  1973-02-01                            360.900
## 3  1973-03-01                            400.161
## 4  1973-04-01                            380.470
## 5  1973-05-01                            392.141
## 6  1973-06-01                            377.232
## 7  1973-07-01                            367.325
## 8  1973-08-01                            353.757
## 9  1973-09-01                            307.006
## 10 1973-10-01                            323.453
```

Stochastic Trend and Stationarity Tests

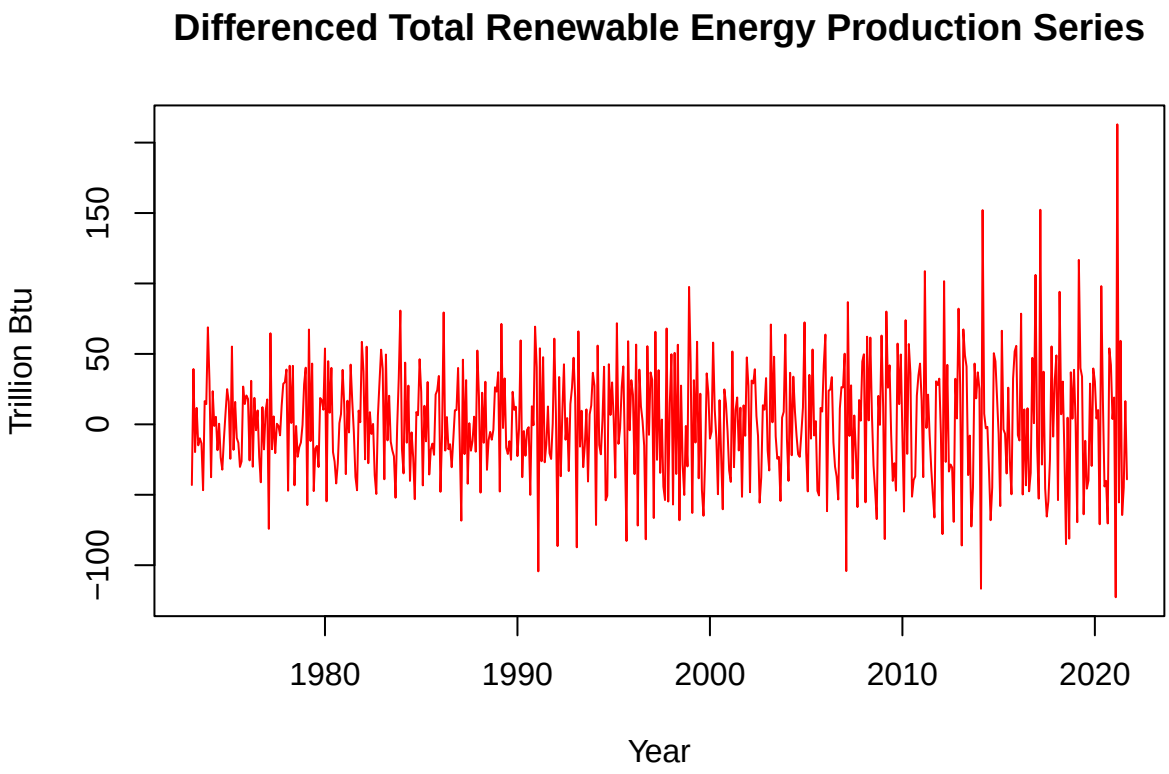
Q1

Difference the "Total Renewable Energy Production" series using function diff(). Function diff() is from package base and take three main arguments: * *x* vector containing values to be differenced; * *lag* integer indicating with lag to use; * *differences* integer indicating how many times series should be differenced.

Try differencing at lag 1 only once, i.e., make `lag=1` and `differences=1`. Plot the differenced series Do the series still seem to have trend?

After differencing the series, they no longer seem to have a trend. The series look to be centered around zero.

```
tsA4 <- ts(data = df$`Total Renewable Energy Production`, start = 1973, frequency = 12)
diffPlot <- diff(tsA4,lag=1,differences=1)
plot(diffPlot, main="Differenced Total Renewable Energy Production Series", ylab="Trillion Btu", col="r
```



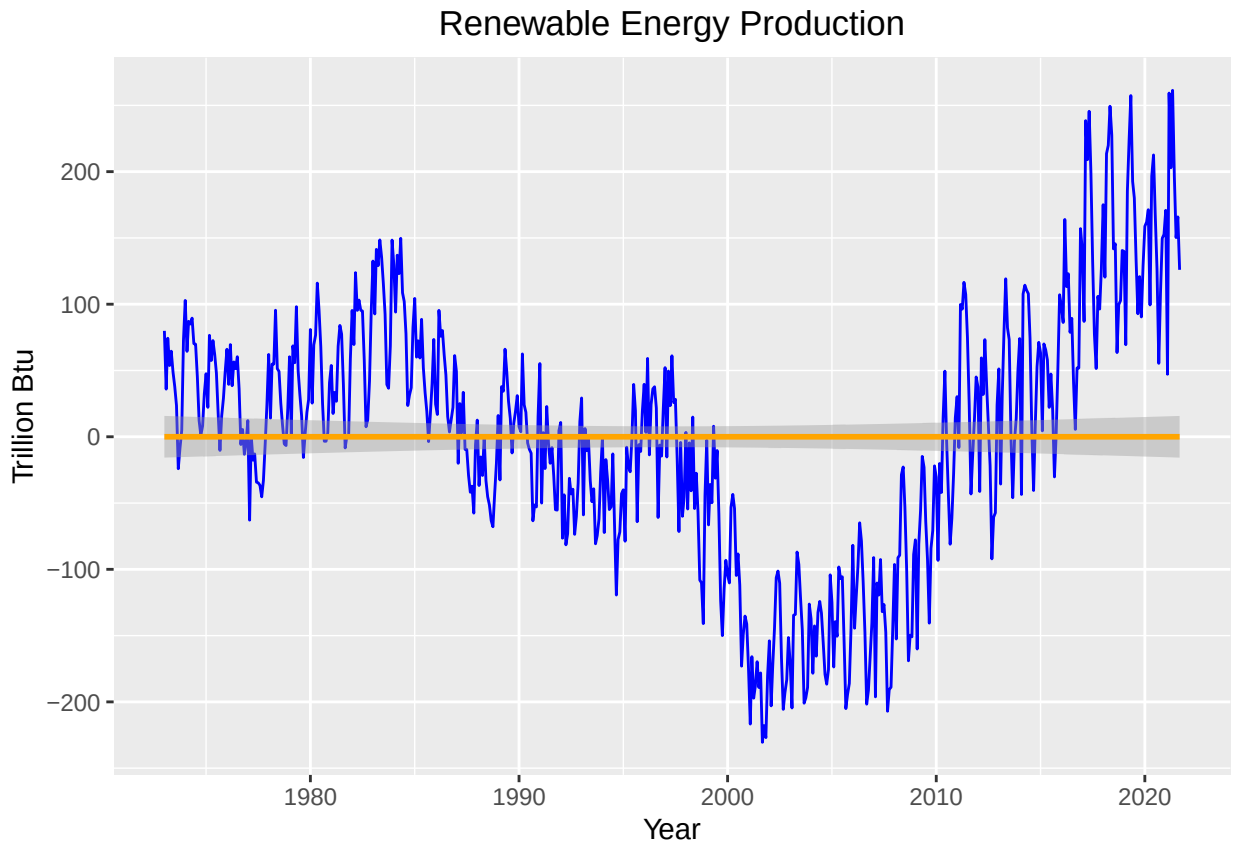
Q2

Now let's compare the differenced series with the detrended series you calculated on A3. In other words, for the "Total Renewable Energy Production" compare the differenced series from Q1 with the series you detrended in A3 using linear regression. (Hint: Just copy and paste part of your code for A3)

Copy and paste part of your code for A3 where you compute regression for Total Energy Production and the detrended Total Energy Production

```
nobs = nrow(df)
t<-c(1:nobs)
lmRE = lm(df$`Total Renewable Energy Production`~t)
b0RE=as.numeric(lmRE$coefficients[1])
b1RE=as.numeric(lmRE$coefficients[2])
detrend_RE <- df[,2]-(b0RE+b1RE*t)
detrend_RE <- as.data.frame(detrend_RE)
ggplot(df, aes(x=df$Month, y=detrend_RE$detrend_RE)) +
            geom_line(color="blue") +
            ylab(paste0(colnames(df))) +
            #geom_line(aes(y=detrend_RE$detrend_RE), color="green") +
            geom_smooth(aes(y=detrend_RE$detrend_RE),color="orange",method="lm") +
            ggtitle("Renewable Energy Production")+
            theme(plot.title = element_text(hjust = 0.5))+  xlab("Year") + ylab("Trillion Btu")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Q3

Create a data frame with 4 columns: month, original series, detrended by Regression Series and differenced series. Make sure you properly name all columns. Also note that the differenced series will have only 584 rows because you loose the first observation when differencing. Therefore, you need to remove the first observations for the original series and the detrended by regression series to build the new data frame.

```
dfQ3 <- df[2:585,1:2] #dropping first row off the df
detrend_RE_Q3 <- detrend_RE[2:585,] #dropping first row off detrended series
dfQ3 <- cbind(dfQ3, "Detrended Series"=detrend_RE_Q3,"Differenced Series"=diffPlot) #combining into one
head(dfQ3,10)
```

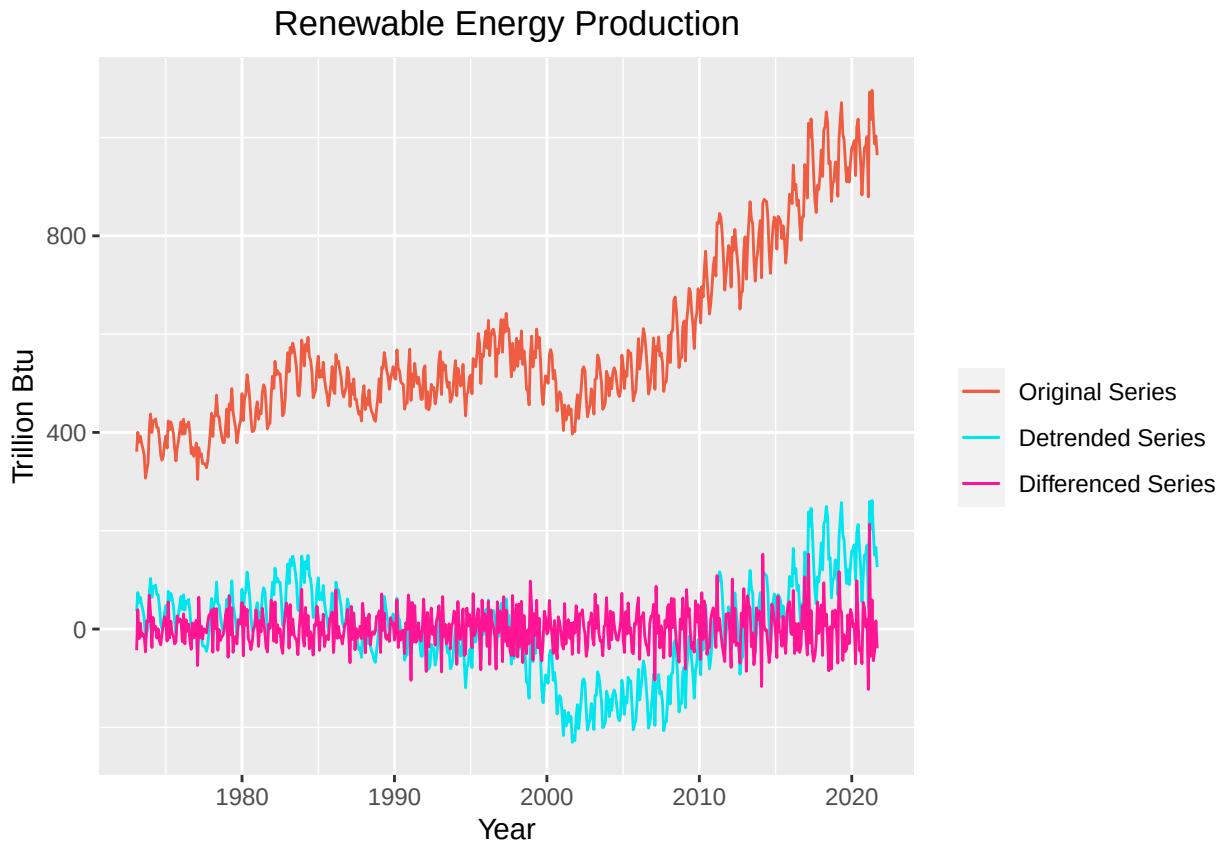
```
##         Month Total Renewable Energy Production Detrended Series
## 2  1973-02-01                            360.900        35.956552
## 3  1973-03-01                            400.161        74.337046
## 4  1973-04-01                            380.470        53.765539
## 5  1973-05-01                            392.141        64.556033
## 6  1973-06-01                            377.232        48.766526
## 7  1973-07-01                            367.325        37.979020
## 8  1973-08-01                            353.757        23.530514
## 9  1973-09-01                            307.006       -24.100993
## 10 1973-10-01                            323.453        -8.534499
## 11 1973-11-01                            337.817         4.948994
##    Differenced Series
## 2             -43.081
```

```
## 3              39.261
## 4             -19.691
## 5              11.671
## 6             -14.909
## 7              -9.907
## 8             -13.568
## 9             -46.751
## 10             16.447
## 11             14.364
```

Q4

Using ggplot() create a line plot that shows the three series together. Make sure you add a legend to the plot.

```
#Use ggplot
ggplot(data = dfQ3, aes(x = dfQ3$Month)) +
            ylab("Trillion Btu") +
            xlab("Year") +
            geom_line(aes(y = dfQ3$`Total Renewable Energy Production`, color="Original Series")) +
            geom_line(aes(y=dfQ3$`Detrended Series`, color="Detrended Series")) +
            geom_line(aes(y=dfQ3$`Differenced Series`, color="Differenced Series")) +
            ggtitle("Renewable Energy Production") +
            theme(plot.title = element_text(hjust = 0.5))+  xlab("Year") + ylab("Trillion Btu") +
            scale_colour_manual("",
                      breaks = c("Original Series", "Detrended Series", "Differenced Series"),
                      values = c("tomato2", "turquoise2", "deeppink1"))
```

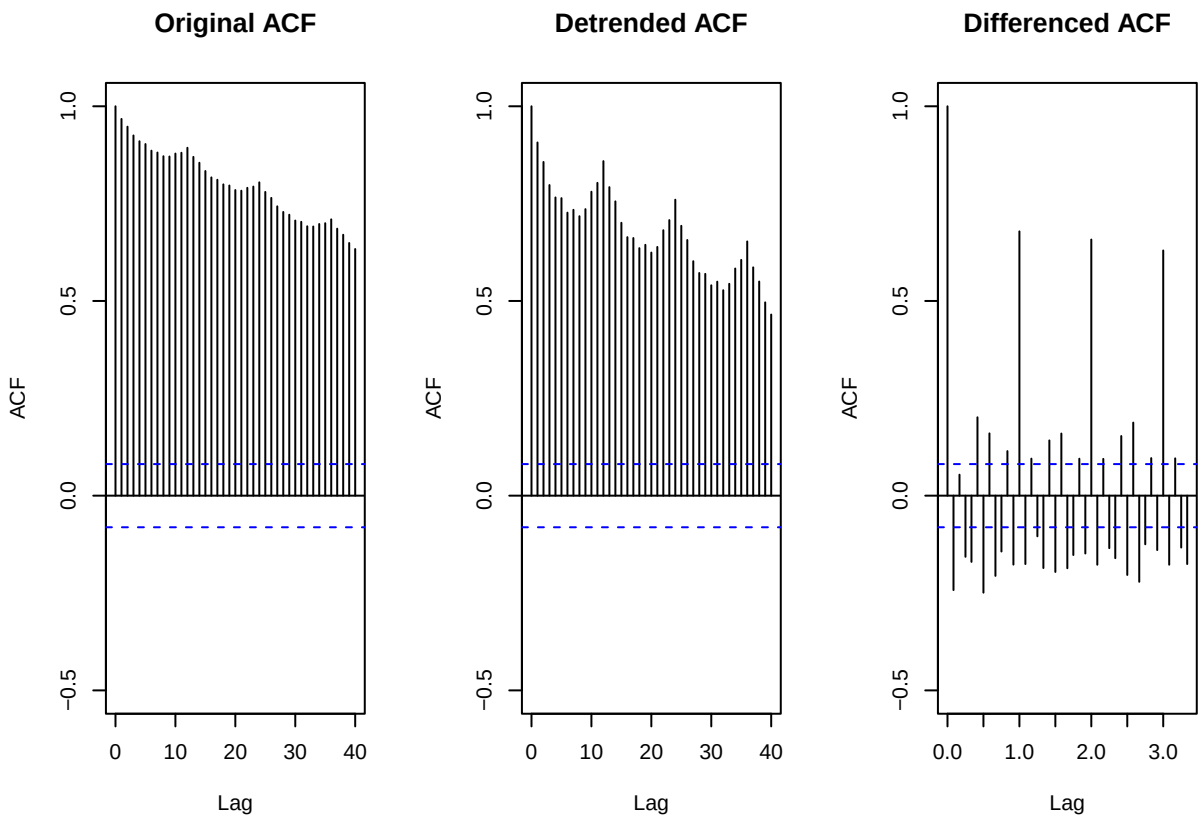


Q5

Plot the ACF for the three series and compare the plots. Add the argument `ylim=c(-0.5,1)` to the Acf() function to make sure all three y axis have the same limits. Which method do you think was more efficient in eliminating the trend? The linear regression or differencing?

The differencing was much more efficient at eliminating the trend. This is evidenced by the fact that the ACF values are much lower than that in the detrended ACF, if not completely insignificant. In fact, the only lags that showed any auto-correlation were multiples of 12 which shows a degree of seasonality remains. but even these were slightly lower than in the detrended ACF.

```
par(mfrow = c(1,3))
acf(dfQ3$`Total Renewable Energy Production`, lag.max = 40, type = "correlation", plot=TRUE, main="Orig
acf(dfQ3$`Detrended Series`, lag.max = 40, type = "correlation", plot=TRUE, main="Detrended ACF", ylim =
acf(dfQ3$`Differenced Series`, lag.max = 40, type = "correlation", plot=TRUE, main="Differenced ACF", yl
```



Q6

Compute the Seasonal Mann-Kendall and ADF Test for the original "Total Renewable Energy Production" series. Ask R to print the results. Interpret the results for both test. Whats the conclusion from the Seasonal Mann Kendall test? What's the conclusion for the ADF test? Do they match what you observed in Q2? Recall that having a unit root means the series has a stochastic trend. And when a series has stochastic trend we need to use a different procedure to remove the trend.

The seasonal Mann Kendall test has a very small p-value, below the standard significance level of .05. This low p value indicates a rejection of the null hypothesis that the total renewable energy production in stationary, in support of it not being stationary.

The ADF test signifies that the time series has a unit root because it has a phi value of -1.4383 that is less than 1.

These results match the observation from Q2 because if the time series were stationary and not unit root, the detrended plot would have stayed close to zero and any deviations would have seemed random. But in reality, the graph of the detrended series varied from zero widely and showed that it did not closely follow a single trend line.

These results mean we have to use a different procedure to remove the trend than simply running a linear regression and using the resulting coefficients to detrend the data.

```
#seasonal mann kendall
SMKtest <- SeasonalMannKendall(tsA4)
print("Results for Seasonal Mann Kendall")
```

```
## [1] "Results for Seasonal Mann Kendall"
```

```
print(summary(SMKtest))
```

```
## Score =  9984 , Var(Score) = 159104
## denominator =  13968
## tau = 0.715, 2-sided pvalue =< 2.22e-16
## NULL
```

```
#augmented dickey fuller
print("Results for ADF test")
```

```
## [1] "Results for ADF test"
```

```
print(adf.test(tsA4,alternative = "stationary"))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  tsA4
## Dickey-Fuller = -1.4383, Lag order = 8, p-value = 0.8161
## alternative hypothesis: stationary
```

Q7

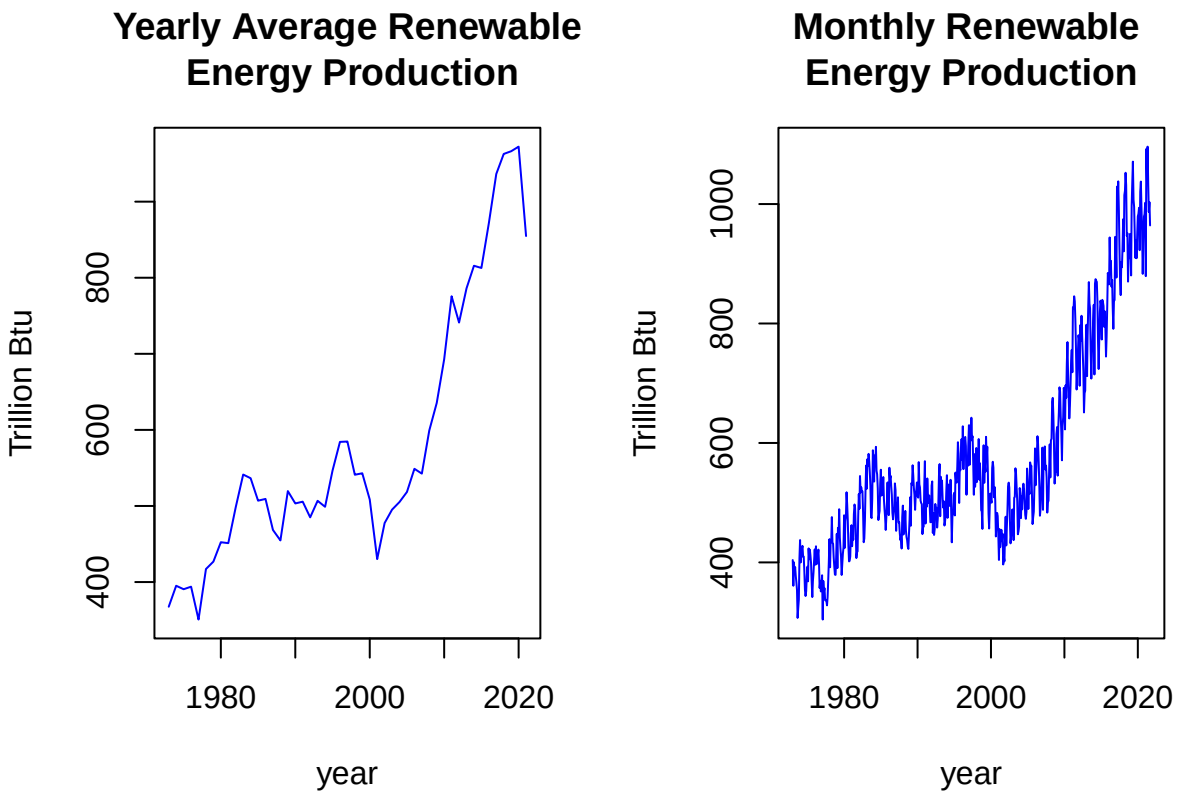
Aggregate the original "Total Renewable Energy Production" series by year. You can use the same procedure we used in class. Store series in a matrix where rows represent months and columns represent years. And then take the columns mean using function colMeans(). Recall the goal is the remove the seasonal variation from the series to check for trend.

```
seasonalMatrix <- matrix(tsA4,byrow=FALSE,nrow=12)
```

```
## Warning in matrix(tsA4, byrow = FALSE, nrow = 12): data length [585] is not a
## sub-multiple or multiple of the number of rows [12]
```

```
yearlyMeans <- colMeans(seasonalMatrix)

#plotting yearly means and original data side by side
year <- format(df$Month, format = "%Y")
year <- as.numeric(unique(year))
par(mfrow = c(1,2))
plot(year, yearlyMeans, ylab = "Trillion Btu", type = "l", col = "blue", main = "Yearly Average Renewabl
plot(df$Month,df$`Total Renewable Energy Production`, type = "l", col = "blue", main = "Monthly Renewabl
```



Q8

Apply the Mann Kendal, Spearman correlation rank test and ADF. Are the results from the test in agreement with the test results for the non-aggregated series, i.e., results for Q6?

The results from the test are in agreement with the test results for the non-aggregated series. In fact, the tau and p-value results for the Mann Kendall test were identical. The results of all three show that the renewble energy production data follows a unit root, stochastic, non-stationary trend.

```
print("Results for Mann Kendall Test")
```

```
## [1] "Results for Mann Kendall Test"
```

```
print(summary(MannKendall(yearlyMeans)))
```

```
## Score =  854 , Var(Score) = 13458.67
## denominator =  1176
## tau = 0.726, 2-sided pvalue =< 2.22e-16
## NULL
```

```
print("Results from Spearman Correlation")
```

```
## [1] "Results from Spearman Correlation"
```

```
sp_rho=cor.test(yearlyMeans,year,method="spearman")
print(sp_rho)
```

```
##
##  Spearman's rank correlation rho
##
## data:  yearlyMeans and year
## S = 2578, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.8684694
```

```
#augmented dickey fuller
print("Results for ADF test")
```

```
## [1] "Results for ADF test"
```

```
print(adf.test(yearlyMeans,alternative = "stationary"))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  yearlyMeans
## Dickey-Fuller = -2.2085, Lag order = 3, p-value = 0.4907
## alternative hypothesis: stationary
```