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

Assignment 4 - Due date 02/12/24

Jenn McNeill

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 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_A04_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 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.\

```
#Load/install required package here
library(readxl)
library(ggplot2)
library(forecast)
```

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

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

```
##
## Attaching package: 'lubridate'

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

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".

```r
#Import the dataset
renewable_energy_raw <- 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`
```

```r
#Extract the column names from row 11
read_col_names <- read_excel(path="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Sou
```

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

```
## * `` -> `...13`
## * `` -> `...14`
```

```
#Add column names to the table
colnames(renewable_energy_raw) <- read_col_names

#Check column names
head(renewable_energy_raw)
```

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

```
#Select columns of interest
renewables <- select(renewable_energy_raw, `Month`, `Total Renewable Energy Production`)

#Make sure that the last year of data is complete
tail(renewables)
```

```
## # A tibble: 6 x 2
##   Month               `Total Renewable Energy Production`
##   <dttm>                                            <dbl>
## 1 2023-04-01 00:00:00                                700.
## 2 2023-05-01 00:00:00                                741.
## 3 2023-06-01 00:00:00                                692.
## 4 2023-07-01 00:00:00                                712.
## 5 2023-08-01 00:00:00                                712.
## 6 2023-09-01 00:00:00                                666.
```

```
#Change indexing so that 9 months of 2023 are removed
renewables <- renewables[1:600,]

#Confirm that the data ends at December 2022
tail(renewables)
```

```
## # A tibble: 6 x 2
##   Month               `Total Renewable Energy Production`
##   <dttm>                                            <dbl>
## 1 2022-07-01 00:00:00                                713.
```

```
## 2 2022-08-01 00:00:00                                672.
## 3 2022-09-01 00:00:00                                633.
## 4 2022-10-01 00:00:00                                659.
## 5 2022-11-01 00:00:00                                686.
## 6 2022-12-01 00:00:00                                680.
```

```r
#Change the date to a date object
Date <- ymd(renewables$Month)

#Add the newly formatted date to a new dataframe
renewables <- cbind(Date,renewables[,2])

#Create a time series object
ts_rp <- ts(renewables[,2],start=c(1973,1), frequency=12)
```

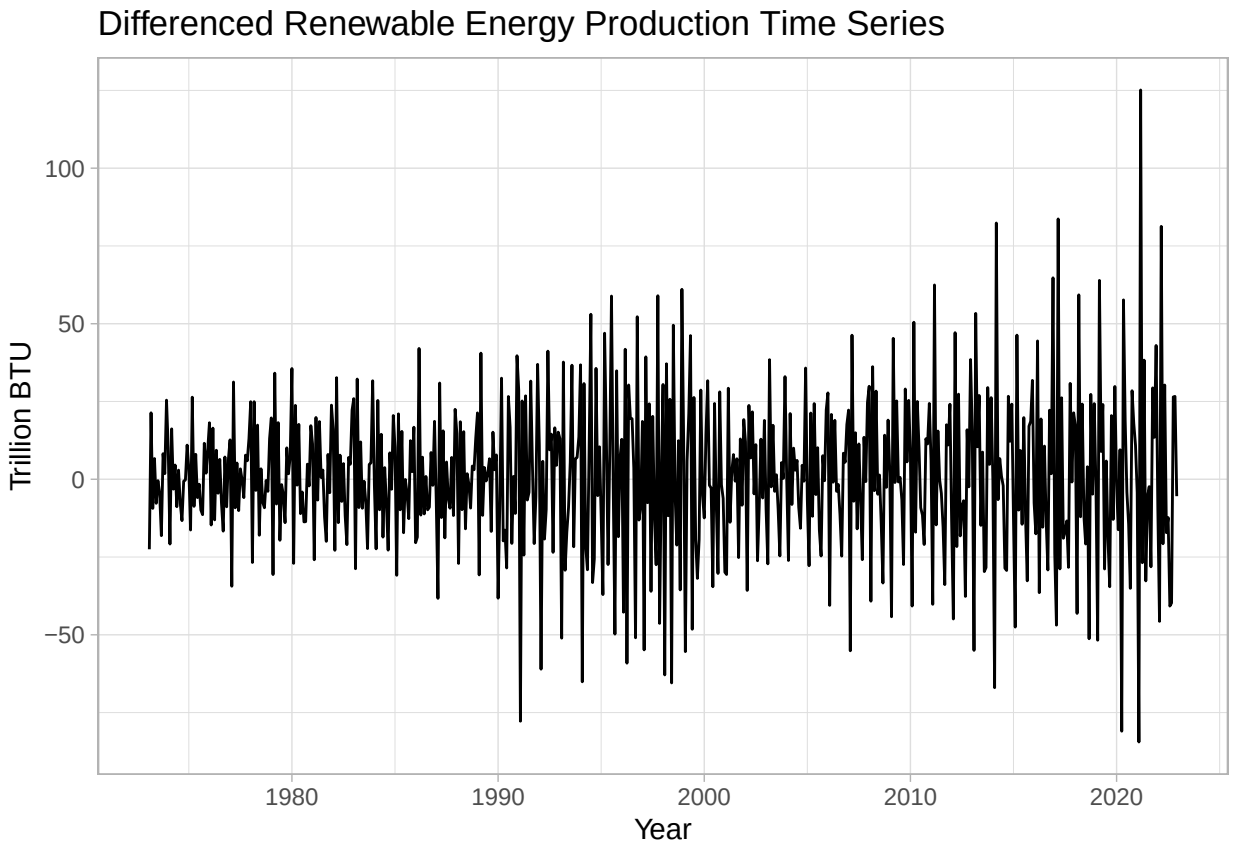
Stochastic Trend and Stationarity Tests

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?

```r
ts_rp_diff <- diff(ts_rp, lag=1, differences=1)

autoplot(ts_rp_diff)+
  xlab("Year")+
  ylab("Trillion BTU")+
  ggtitle("Differenced Renewable Energy Production Time Series")+
  theme_light()
```



Differencing the series seems to remove the trend. There are still fluctuations in the amplitude of the plot, but the slope is not visibly positive or negative.

Q2

Copy and paste part of your code for A3 where you run the regression for Total Renewable Energy Production and subtract that from the original series. This should be the code for Q3 and Q4. make sure you use the same name for you time series object that you had in A3.

```
#Create variables to use for indexing
nobs <- nrow(renewables)
t <- 1:nobs

#Fit a linear trend to the renewable production time series
rp_lm <- lm(renewables[,2]~t)
summary(rp_lm)
```

```
##
## Call:
## lm(formula = renewables[, 2] ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -146.12  -34.41   10.85   39.97  149.93
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 183.91273    4.87034   37.76   <2e-16 ***
## t             0.68956    0.01404   49.11   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 59.57 on 598 degrees of freedom
## Multiple R-squared:  0.8013, Adjusted R-squared:  0.801
## F-statistic:  2412 on 1 and 598 DF,  p-value: < 2.2e-16
```

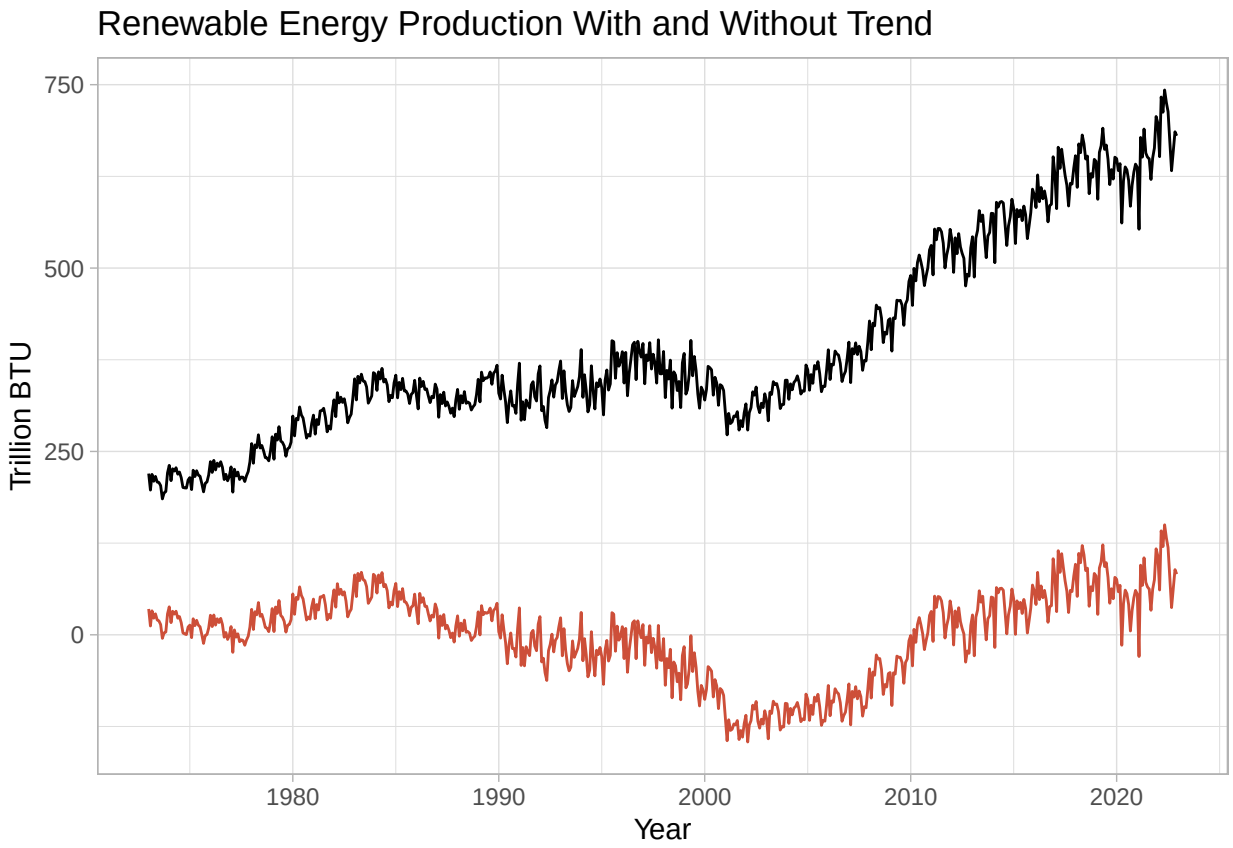
```
#Store slope and intercept coefficients
rp_beta0 <- rp_lm$coefficients[1]
rp_beta1 <- rp_lm$coefficients[2]

#Print the coefficients
print(rp_beta0)
```

```
## (Intercept)
##    183.9127
```

```
print(rp_beta1)
```

```
##         t
## 0.6895634
```

```
#Store the detrended renewable production series
rp_detrend <- renewables[,2] - (rp_beta0 + (rp_beta1*t))

#Plot
ggplot(renewables, aes(x=Date))+
  geom_line(aes(y=renewables[,2]), color = "black")+
  geom_line(aes(y=rp_detrend), color = "tomato3")+
  ggtitle("Renewable Energy Production With and Without Trend")+
  xlab("Year")+
  ylab("Trillion BTU")+
  theme_light()
```

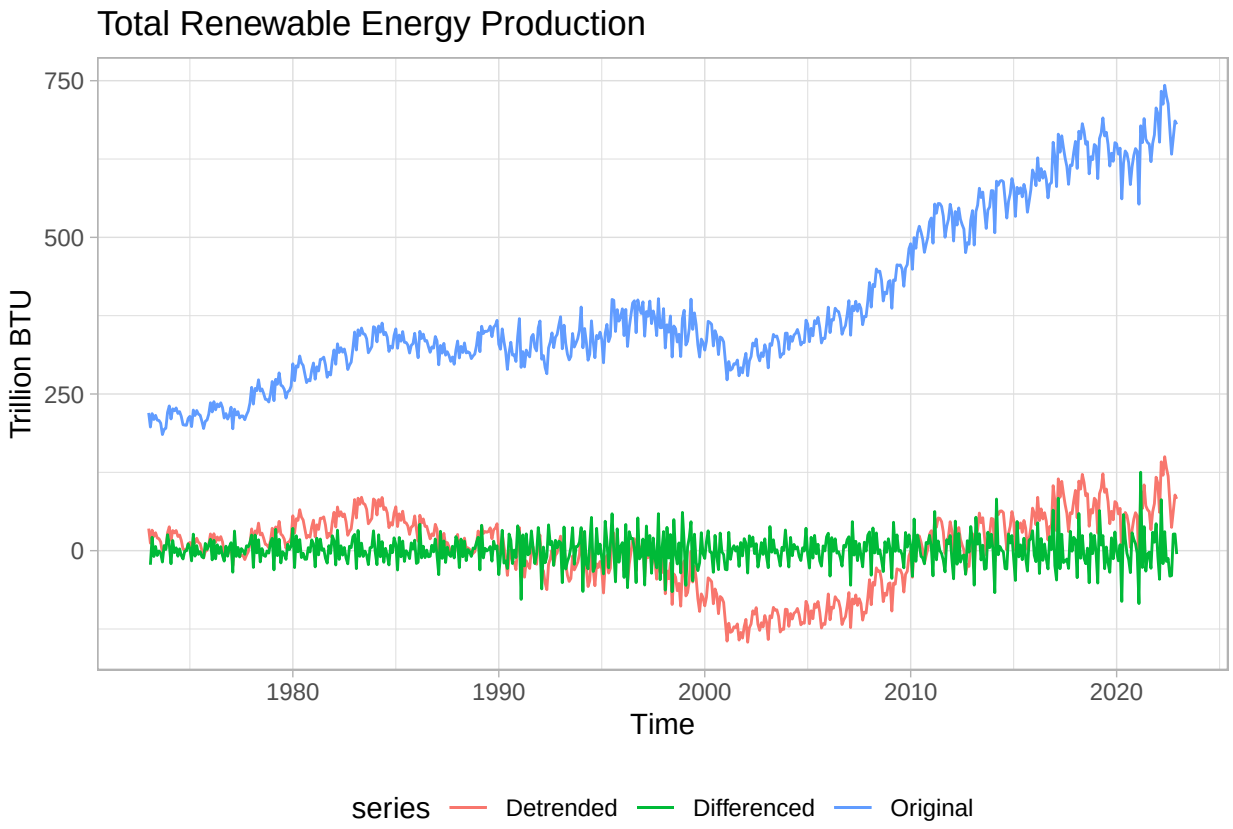


Q3

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 Q2 using linear regression.

Using autoplot() + autolayer() create a plot that shows the three series together. Make sure your plot has a legend. The easiest way to do it is by adding the `series=` argument to each autoplot and autolayer function. Look at the key for A03 for an example.

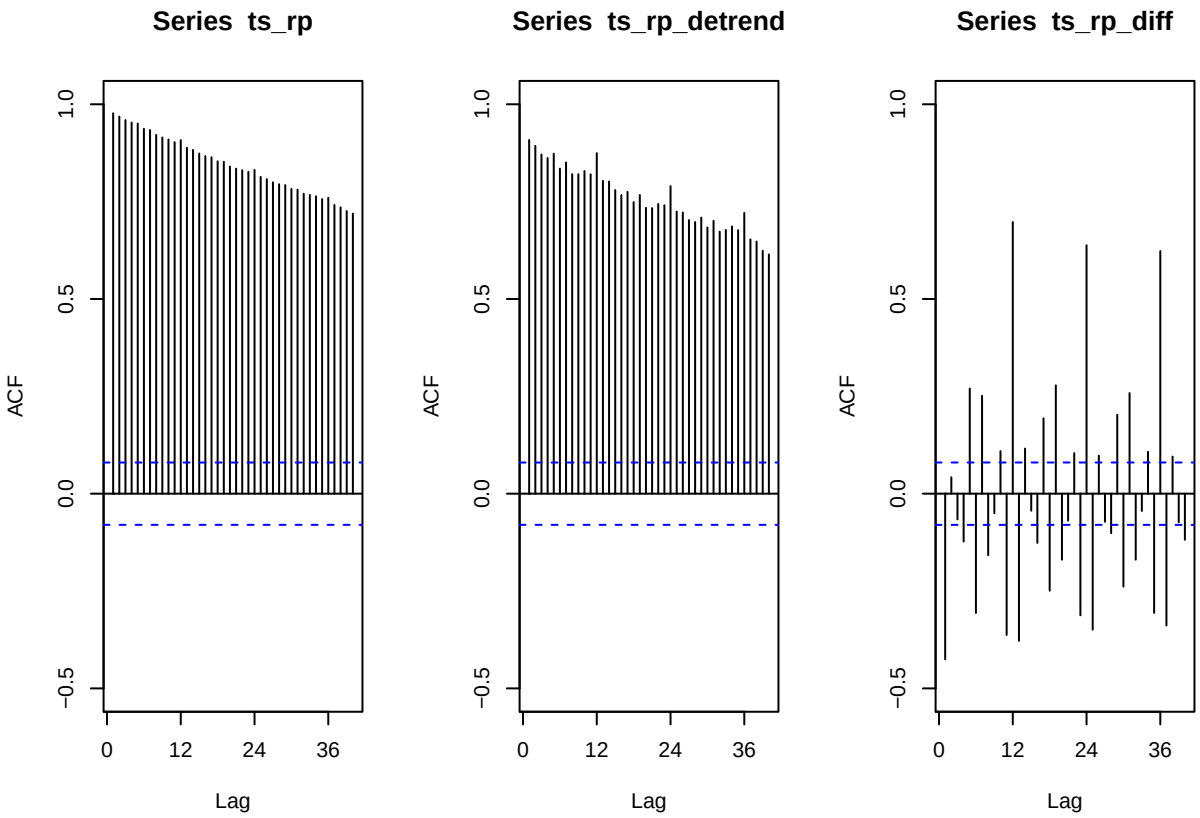
```
#Convert detrended series to a time series object
ts_rp_detrend <- ts(rp_detrend,start=c(1973,1), frequency=12)

#Plot all three time series objects
autoplot(ts_rp,series="Original")+
  autolayer(ts_rp_detrend,series="Detrended")+
  autolayer(ts_rp_diff,series="Differenced")+
  ylab("Trillion BTU")+
  ggtitle("Total Renewable Energy Production")+
  theme_light()+
  theme(legend.position = "bottom")
```



Q4

Plot the ACF for the three series and compare the plots. Add the argument `ylim=c(-0.5,1)` to the autoplot() or Acf() function - whichever you are using to generate the plots - 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?

```
#Compare the Acf for renewable production with and without trend
par(mfrow=c(1,3))
  Acf(ts_rp,lag.max=40,ylim=c(-0.5,1),plot=TRUE)
  Acf(ts_rp_detrend,lag.max=40,ylim=c(-0.5,1),plot=TRUE)
  Acf(ts_rp_diff,lag.max=40,ylim=c(-0.5,1),plot=TRUE)
```



I think that differencing was more efficient in eliminating the trend. The ACFs for the original time series and the detrended time series both have the same shape, which means that the detrended series' values still rely heavily on the values at one lag prior. With the differenced time series, the ACF shows that lags 12, 24, and 36 have the highest autocorrelation and that the value at a given time relies less on the time step one before it. Additionally, the maximum magnitude of the autocorrelation for the differenced time series is close to the minimum magnitude of the autocorrelation for the other two time series. Differencing the series lowers the overall autocorrelation and removes the trend more efficieintly than performing the linear regression.

Q5

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. What is 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.

```
SMK_ts_rp <- SeasonalMannKendall(ts_rp)
print(summary(SMK_ts_rp))
```

```
## Score =  11423 , Var(Score) = 171499
## denominator =  14699.5
## tau = 0.777, 2-sided pvalue =< 2.22e-16
## NULL
```

```
ADF_ts_rp <- adf.test(ts_rp,alternative="stationary")
print(ADF_ts_rp)
```

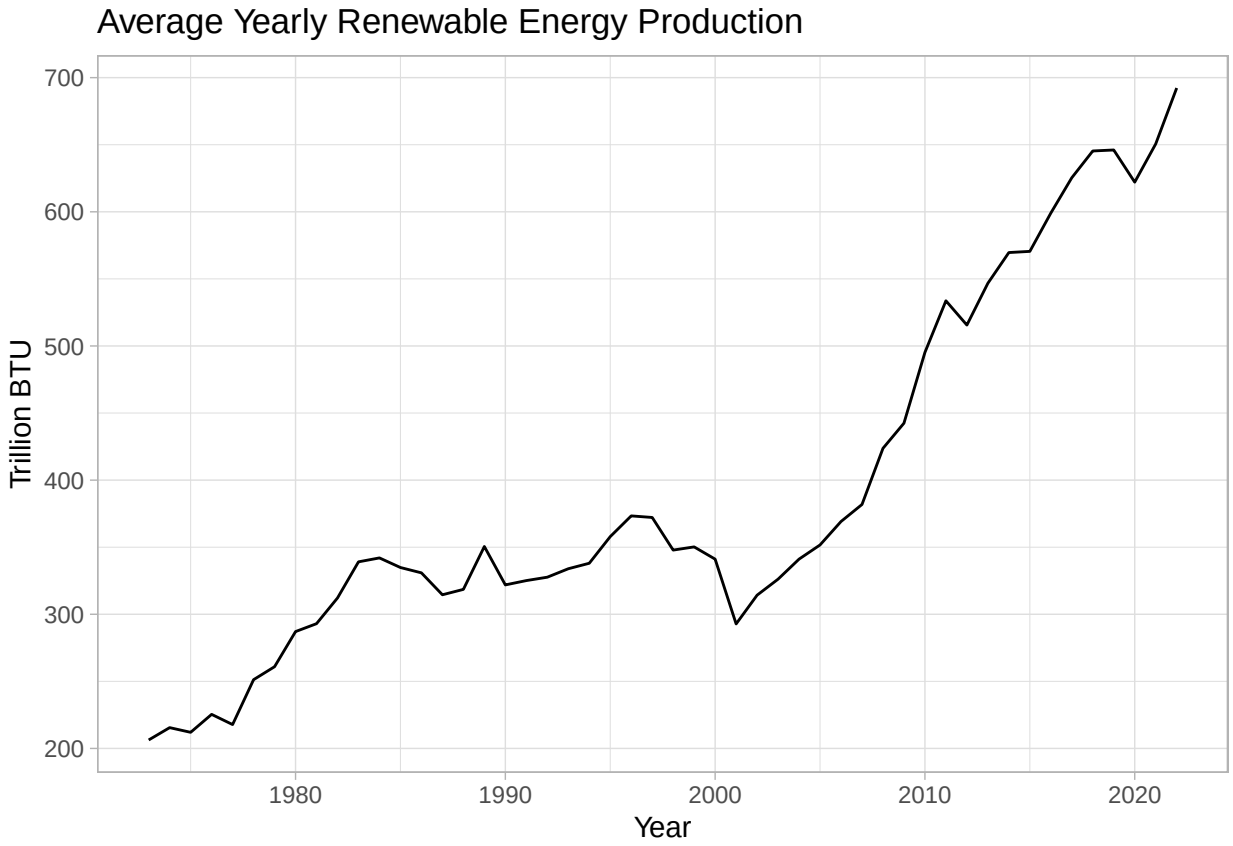
```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_rp
## Dickey-Fuller = -1.3129, Lag order = 8, p-value = 0.8691
## alternative hypothesis: stationary
```

The results of the Seasonal Mann Kendall show that there is a trend in the data because the p-value is 2.22e-16, which is less than .05. According to this p-value, we can reject the null hypothesis that there is no trend. Additionally, the score is positive, which means that the trend is increasing. This matches what I observed in Q2, which is that the original time series had a positive trend. The ADF results in a p-value of 0.87, which is above .05. This means we fail to reject the null hypothesis that there is a unit root, and we can assume that our data has a stochastic trend. This matches what we said in Q2 because the trend was removed by differencing and not by the linear detrending method.

Q6

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. Convert the accumulates yearly series into a time series object and plot the series using autoplot().

```
rp_matrix <- matrix(ts_rp,byrow=FALSE,nrow=12)
rp_yearly <- colMeans(rp_matrix)

years <- c(year(first(Date)):year(last(Date)))

rp_yearly <- data.frame(years, rp_yearly)

ts_rp_yearly <- ts(rp_yearly[,2],start=c(1973,1))

autoplot(ts_rp_yearly)+
  xlab("Year")+
  ylab("Trillion BTU")+
  ggtitle("Average Yearly Renewable Energy Production")+
  theme_light()
```



Q7

Apply the Mann Kendall, Spearman correlation rank test and ADF. Are the results from the test in agreement with the test results for the monthly series, i.e., results for Q6?

```
MK_ts_rp_yearly <- MannKendall(ts_rp_yearly)
print(summary(MK_ts_rp_yearly))
```

```
## Score =  983 , Var(Score) = 14291.67
## denominator =  1225
## tau = 0.802, 2-sided pvalue =< 2.22e-16
## NULL
```

```
Sp_rho_ts_rp_yearly <- cor.test(ts_rp_yearly,years,method="spearman")
print(Sp_rho_ts_rp_yearly)
```

```
##
##  Spearman's rank correlation rho
##
## data:  ts_rp_yearly and years
## S = 1852, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9110684
```

```r
ADF_ts_rp_yearly <- adf.test(ts_rp_yearly,alternative="stationary")
print(ADF_ts_rp_yearly)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_rp_yearly
## Dickey-Fuller = -1.0881, Lag order = 3, p-value = 0.9156
## alternative hypothesis: stationary
```

The results when I run the tests on a series of monthly averages are consistent with my results from when I ran the tests on the original time series. The Mann-Kendall test and Spearman Rank Correlation tests both provide p-values less than 0.05. Because of these values, I can reject the null hypotheses that there is no trend. The positive scores show that the trend is increasing upwards. Likewise, the ADF gives me a p-value pf 0.92 (more than 0.05), so I fail to reject the null hypothesis that there is a unit root. This means our trend is stochastic, which makes sense because differencing was the preferential method to remove the trend whereas the linear method did not work effectively.