

ENV 797 - Time Series Analysis for Energy and Environment

Applications | Spring 2026

Assignment 4 - Due date 02/10/26

Lauren Shohan

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_Sp26.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(forecast)
library(tseries)
library(dplyr)
library(lubridate)
library(ggplot2)
library(readxl)
library(openxlsx)
library(Kendall)
library(cowplot)
```

Questions

Consider the same data you used for A3 from the spreadsheet “Table_10.1_Renewable_Energy_Production_and_Consumption”. The data comes from the US Energy Information and Administration and corresponds to the December 2025 Monthly Energy Review. **For this assignment you will work only with the column “Total Renewable Energy Production”.**

```
#Importing data set - you may copy your code from A3
#extract data
```

```

energydata <- read_excel(path=
"/home/guest/TSA_Sp26/Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
skip = 12, sheet="Monthly Data",col_names=FALSE)

#extract columns
energydata_cols <- read_excel(path=
"/home/guest/TSA_Sp26/Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
skip = 10 ,n_max = 1, sheet="Monthly Data",col_names=FALSE)

#assign column names to data
colnames(energydata) <- energydata_cols

#select column i want
energydata_1col <- energydata[,5]

#converting into ts
#starting in 1973 and going monthly, thus freq is 12
ts_energydata <- ts(energydata_1col,start=c(1973,1), frequency=12)

```

Stochastic Trend and Stationarity Tests

For this part you will work only with the column Total Renewable Energy Production.

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?

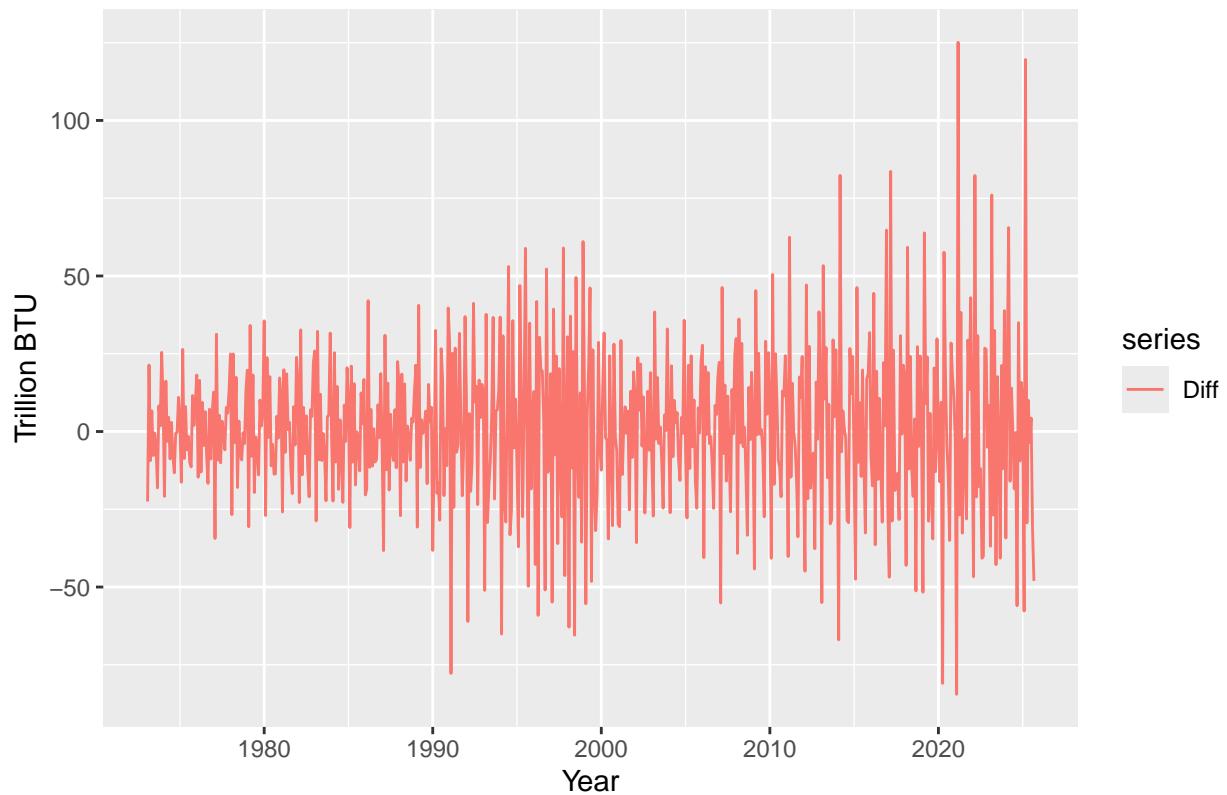
```

diff_renewable <- diff(ts_energydata, lag = 1, differences = 1)
ts_diff_renewable <- ts(diff_renewable, start = c(1973,1), frequency = 12)

autoplot(diff_renewable, series = 'Diff') +
  xlab('Year') +
  ylab('Trillion BTU') +
  ggtitle('Differenced Renewable Production Series')

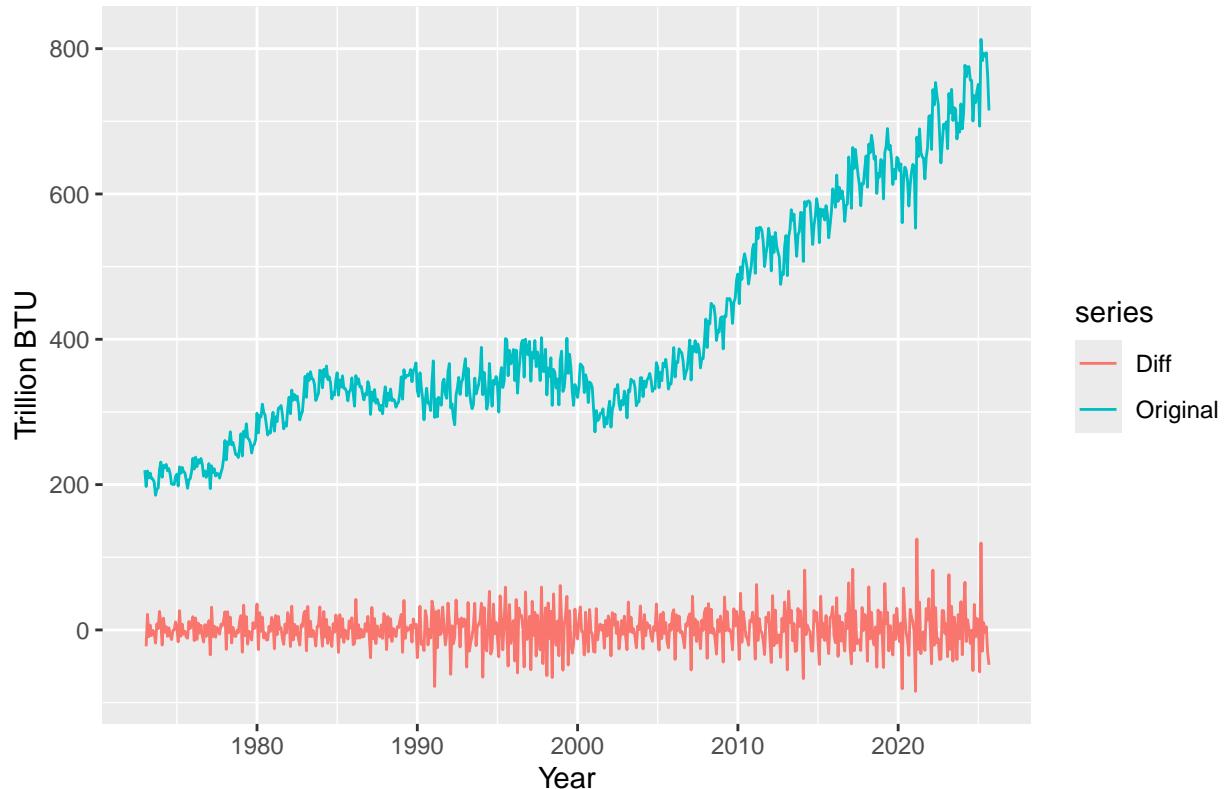
```

Differenced Renewable Production Series



```
autoplot(diff_renewable, series = 'Diff') +  
  autolayer(ts_energydata, series = 'Original') +  
  xlab('Year') +  
  ylab('Trillion BTU') +  
  ggtitle('Differenced vs Original Renewable Production Series')
```

Differenced vs Original Renewable Production Series



> Answer: Since the spikes are oscillating around zero and have an average mean of zero there is no visible trend, it is staying steady. The differenced series compared to the original shows a drastic decrease in the linear trend that was seen in the original series.

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 assign same name for the time series object that you had in A3, otherwise the code will not work.

```
nobs <- nrow(ts_energydata)

#Create vector t
t <- c(1:nobs)

#fit linear trend to ts_renewable and t
lm_renewable <- lm(ts_energydata[,1]~t)
summary(lm_renewable)

## 
## Call:
## lm(formula = ts_energydata[, 1] ~ t)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -154.81  -39.55   12.52   41.49  171.15
```

```

## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 171.44868   5.11085  33.55 <2e-16 ***
## t           0.74999   0.01397  53.69 <2e-16 ***  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 64.22 on 631 degrees of freedom
## Multiple R-squared:  0.8204, Adjusted R-squared:  0.8201 
## F-statistic: 2883 on 1 and 631 DF,  p-value: < 2.2e-16

#Store reg coefficient
beta0_int_renewable <- as.numeric(lm_renewable$coefficients[1]) #intercept
beta1_slope_renewable <- as.numeric(lm_renewable$coefficients[2]) #slope

nobs <- nrow(ts_energydata)

#Create vector t
t <- c(1:nobs)

#fit linear trend to ts renewable and t
lm_renewable <- lm(ts_energydata[,1]~t)
summary(lm_renewable)

## 
## Call:
## lm(formula = ts_energydata[, 1] ~ t)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max
## -154.81  -39.55   12.52   41.49  171.15
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 171.44868   5.11085  33.55 <2e-16 ***
## t           0.74999   0.01397  53.69 <2e-16 ***  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 64.22 on 631 degrees of freedom
## Multiple R-squared:  0.8204, Adjusted R-squared:  0.8201 
## F-statistic: 2883 on 1 and 631 DF,  p-value: < 2.2e-16

#Store reg coefficient
beta0_int_renewable <- as.numeric(lm_renewable$coefficients[1]) #intercept
beta1_slope_renewable <- as.numeric(lm_renewable$coefficients[2]) #slope

#remove the trend from series
detrend_energy_data <- ts_energydata[,1] - (beta0_int_renewable + beta1_slope_renewable*t)
class(detrend_energy_data)

## [1] "ts"

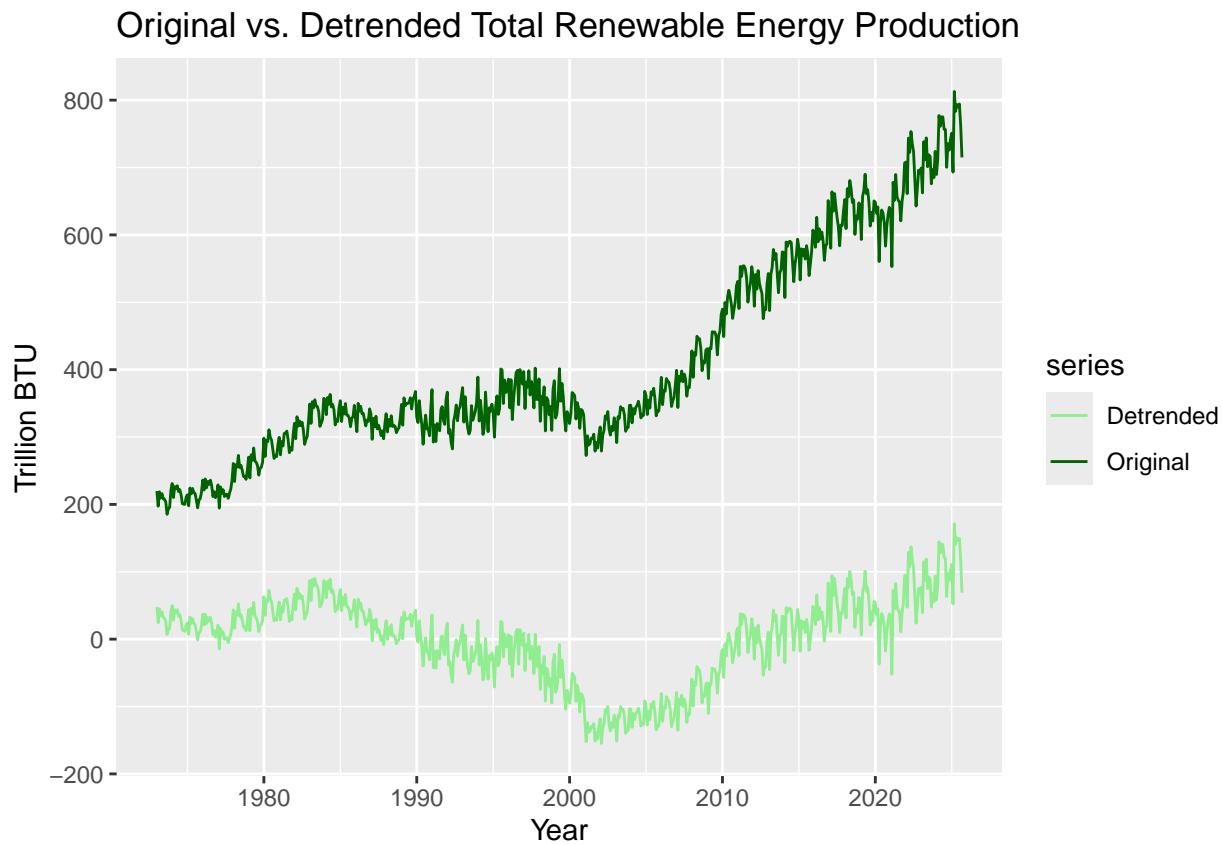
```

```

#transform to ts
ts_detrend_energy <- ts(detrend_energy_data,frequency = 12,start=c(1973,1))

#plot
autoplot(ts_energydata[,1], series = 'Original') +
  autolayer(ts_detrend_energy, series = 'Detrended') +
  xlab("Year") +
  ylab("Trillion BTU") +
  ggtitle('Original vs. Detrended Total Renewable Energy Production') +
  scale_color_manual(values = c('Original' = 'darkgreen', 'Detrended' = 'lightgreen'))

```



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.

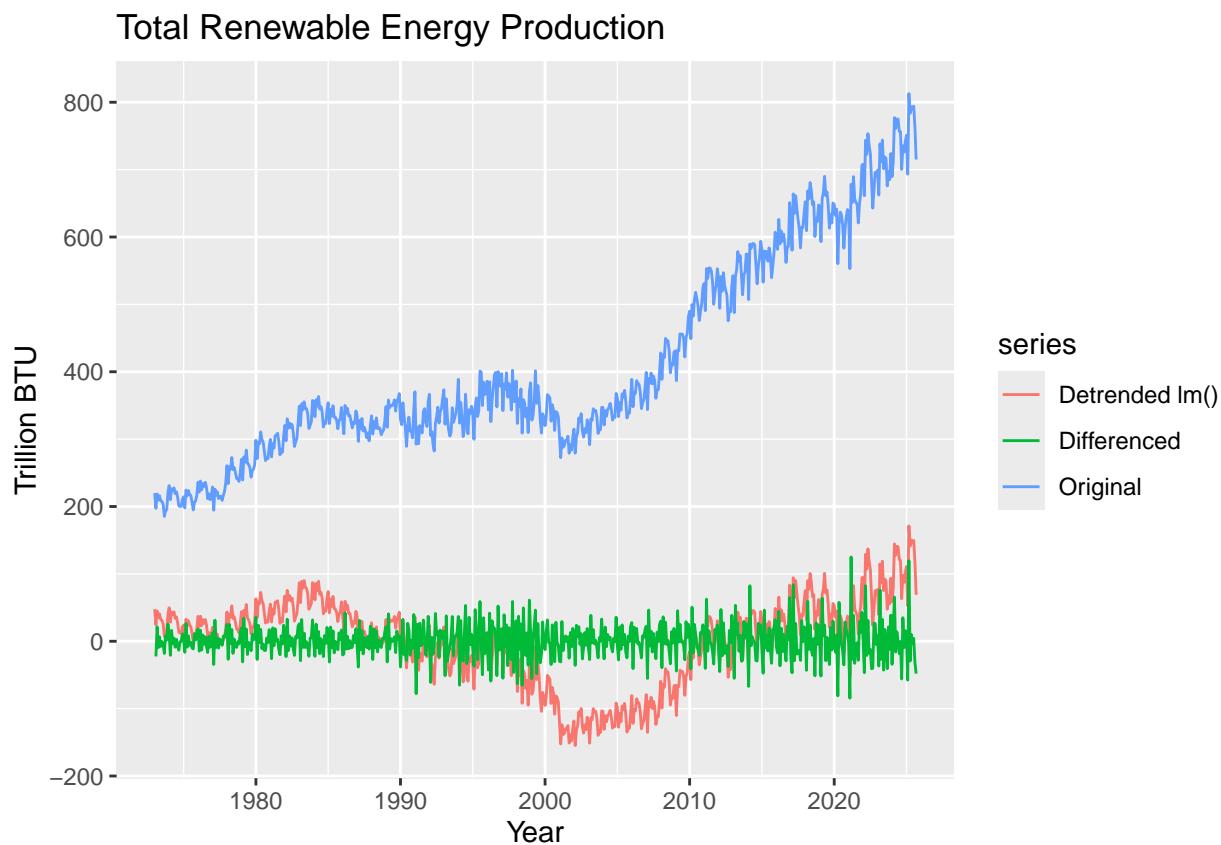
Using `autoplot()` + `autolayer()` create a plot that shows the three series together (i.e. "Original", "Differenced", "Detrended lm()"). Make sure your plot has a legend. The easiest way to do it is by adding the `series=` argument to each `autoplot` and `autolayer` function. Look at the key for A03 for an example on how to use `autoplot()` and `autolayer()`.

What can you tell from this plot? Which method seems to have been more efficient in removing the trend?

```

autoplot(ts_energydata, series = 'Original') +
  autolayer(detrend_energy_data, series = 'Detrended lm()') +
  autolayer(diff_renewable, series = 'Differenced') +
  xlab('Year') +
  ylab('Trillion BTU') +
  ggtitle('Total Renewable Energy Production')

```



Answer: The plot shows that while the detrended series took out a large portion of the upward trend shown in the original and brought it down to zero, the differenced series was more effective. The differenced series fluctuates around zero with no upward or downward trend, showcasing the trend was effectively removed and looks visibly stationary.

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. Looking at the ACF which method do you think was more efficient in eliminating the trend? The linear regression or differencing?

```

plot_grid(
  autoplot(Acf(ts_detrend_energy, lag.max = 40, plot = FALSE)) +
    coord_cartesian(ylim=c(-0.5,1)) + #coord cartesian to make ylim work ! - chat helped with this fix
    ggtitle('ACF Detrended'),
  autoplot(Acf(diff_renewable, lag.max = 40, plot = FALSE)) +

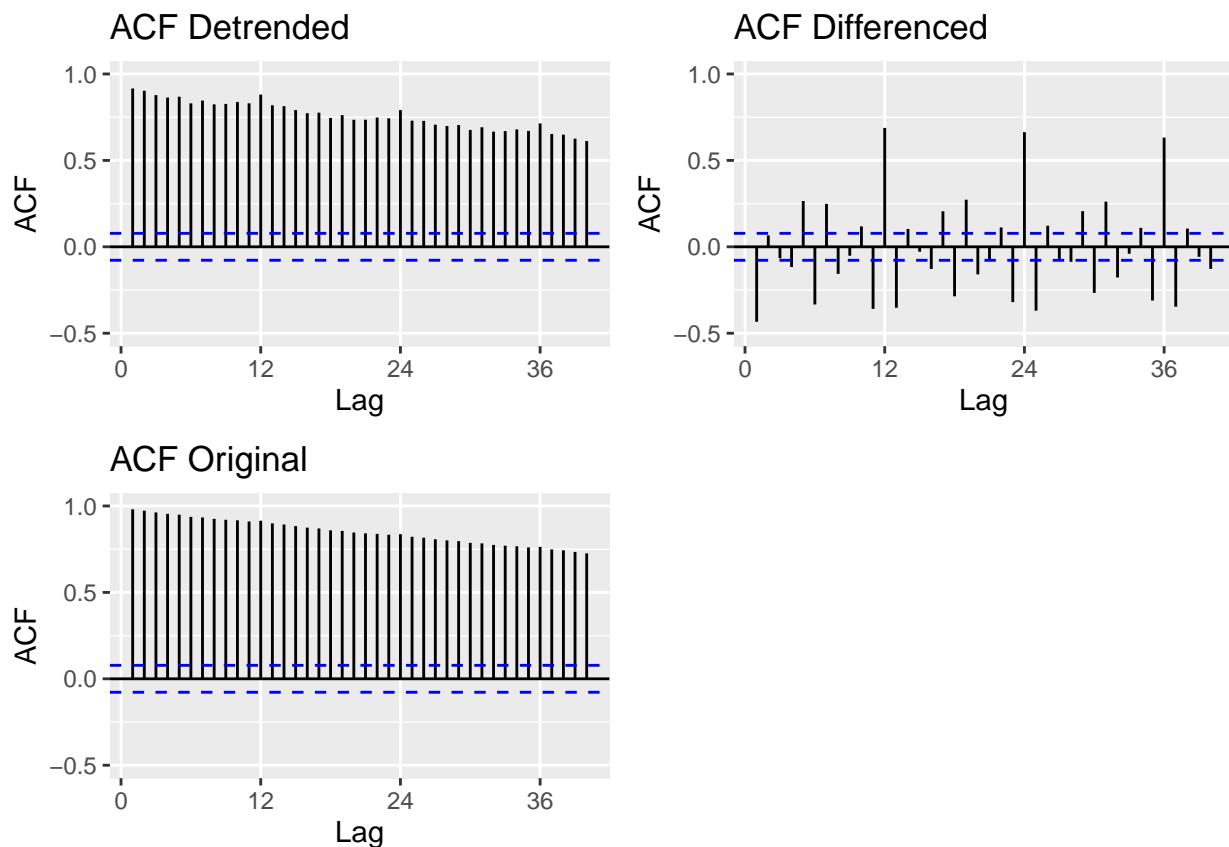
```

```

coord_cartesian(ylim=c(-0.5,1)) +
ggtitle('ACF Detrended'),
autoplot(Acf(ts_energydata, lag.max = 40, plot = FALSE)) +
coord_cartesian(ylim=c(-0.5,1)) +
ggtitle('ACF Differenced')

coord_cartesian(ylim=c(-0.5,1)) +
ggtitle('ACF Original')

```



Answer: Looking at the ACFs, I think the differenced method was more efficient with eliminating the trend and thus showcases the seasonality component of the series. The detrended series looks very similar to the original ACF, with only small spikes at the seasonal months of 12,24,36. However, once differenced you can clearly see the seasonal component with constant spikes at 12,25,36 and there is no trend blocking the seasonality display.

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 Q3 plot? Recall that having a unit root means the series has a stochastic trend. And when a series has stochastic trend we need to use differencing to remove the trend.

```

# ----- mann kendall -----
SMKtest <- SeasonalMannKendall(ts_energydata)
print("Results for Seasonal Mann Kendall /n")

```

```

## [1] "Results for Seasonal Mann Kendall /n"

print(summary(SMKtest))

## Score = 13083 , Var(Score) = 201135
## denominator = 16379.5
## tau = 0.799, 2-sided pvalue =< 2.22e-16
## NULL

# ----- ADF -----
#Null hypothesis is that data has a unit root
print("Results for ADF test/n")

## [1] "Results for ADF test/n"

print(adf.test(ts_energydata,alternative = "stationary"))

## 
##  Augmented Dickey-Fuller Test
##
## data: ts_energydata
## Dickey-Fuller = -1.0247, Lag order = 8, p-value = 0.9347
## alternative hypothesis: stationary

```

Answer: Seasonal Mann Kendall: The results show a tau of 0.799 which indicates a strong positive monotonic trend and the small pvalue of <2.22e-16 means we should reject the null hypothesis that its stationary. The SMK test showcases that there is strong evidence of an increasing total renewable energy production even after tackling the seasonality component and that a trend exists.

ADF test: The ADF test has a dickey-fuller number of -1.0247 and a pvalue of 0.9347, with this large pvalue we do not reject the null hypothesis that the series has a unit root/non-stationary. This showcases total renewable energy production is non-stationary and has a stochastic trend. Yes these results matched what I saw in Q3, with the original series having a strong upward trend and that differencing series removed that trend and fluctuated around zero. Thus these results showcase the need to difference.

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 to remove the seasonal variation from the series to check for trend. Convert the accumulated yearly series into a time series object and plot the series using autoplot().

```

#2025 is not a complete year so i decided to trim it out to do yearly aggregation,
#this will produce similar results if i had also kept those few months

ts_energydata_trimmed <- window(ts_energydata, end=c(2024, 12))

#Group data in yearly steps instances
energy_data_matrix <- matrix(ts_energydata_trimmed,byrow=FALSE,nrow=12)

```

```

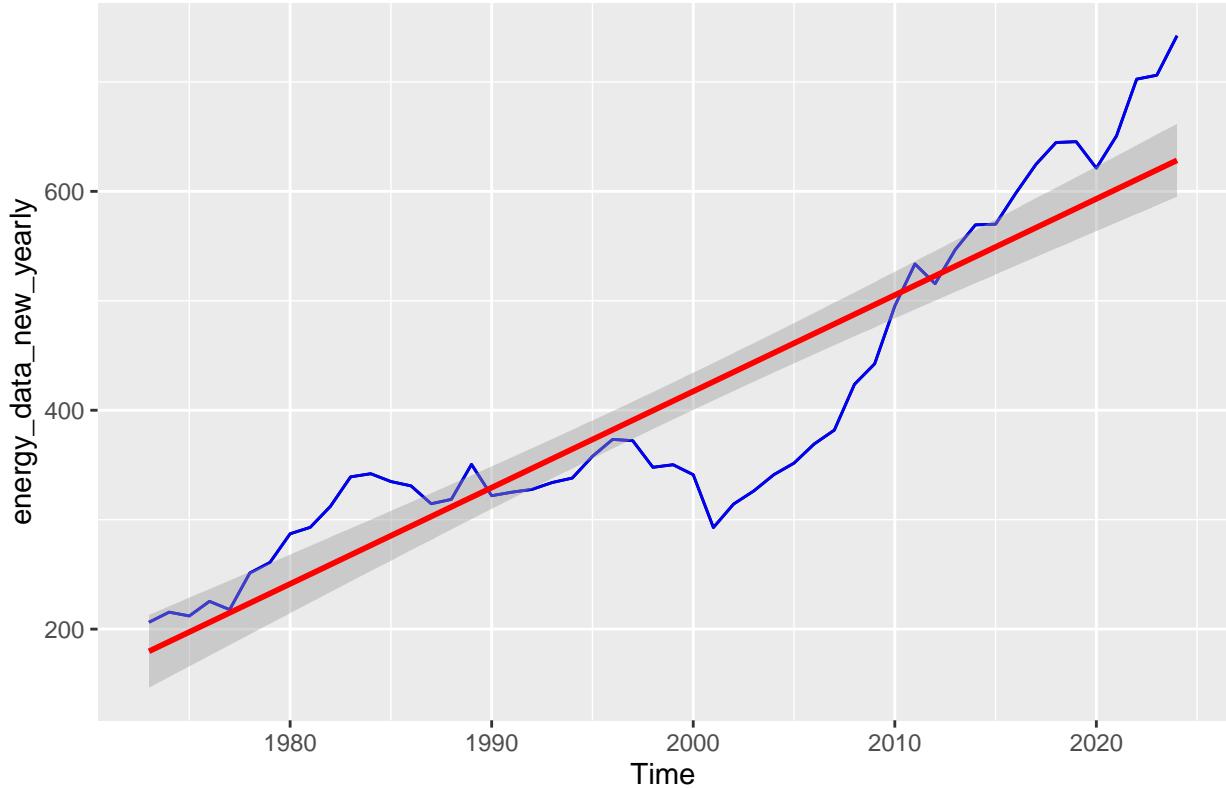
energy_data_yearly <- colMeans(energy_data_matrix)

energy_data_new_yearly <- ts(energy_data_yearly, start = 1973, frequency = 1)

autoplot(energy_data_new_yearly) +
  geom_line(color="blue") +
  geom_smooth(color="red",method="lm")

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

```



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 Q5?

```

# ----- mann kendall -----
SMKtest_yearly <- SeasonalMannKendall(energy_data_new_yearly)
print("Results for Seasonal Mann Kendall Yearly /n")

## [1] "Results for Seasonal Mann Kendall Yearly /n"

```

```

print(summary(SMKtest_yearly))

## Score = 1084 , Var(Score) = 16059.33
## denominator = 1326
## tau = 0.817, 2-sided pvalue <= 2.22e-16
## NULL

# ----- ADF -----
#Null hypothesis is that data has a unit root
print("Results for ADF test/n")

## [1] "Results for ADF test/n"

print(adf.test(energy_data_new_yearly,alternative = "stationary"))

## 
## Augmented Dickey-Fuller Test
##
## data: energy_data_new_yearly
## Dickey-Fuller = -0.85301, Lag order = 3, p-value = 0.9515
## alternative hypothesis: stationary

```

Answer: With the SMK test, the large tau of 0.817 showcases a strong increasing trend and the very small pvalue of <2.22e-16 tells us to reject the null that says there is no trend. This is exactly the same as the results from Q5 monthly results. Thus, there is strong evidence there is a positive monotonic trend in the series. The ADF test of dickey-fuller of -0.853 and pvalue of 0.9515 we fail to reject the null hypothesis and thus this series having a unit root we can conclude the yearly series is non stationary and has a stochastic trend which is what we determined from Q5.