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

Assignment 4 - Due date 02/12/24

Vincient Whatley

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(lubridate)
```

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

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

```
library(ggplot2)
library(forecast)
```

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

```
library(Kendall)
library(tseries)
library(readxl)
library(trend)
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 readxl package

# reading in data with readexcel function

energy_data_raw <-read_excel("Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx
#Triming data for specified colmunis
energydata <- energy_data_raw[,1:5]

#Removing unnecessary columns, renaming columns
energydata <- energydata[, -c(2, 3, 4)]
colnames(energydata)=c("Month",  "TREP")

#convert month day year
energydata$Month <- as.Date(energydata$Month,format = "%Y-%m-%d")

energydata$TREP <- as.numeric(energydata$TREP)
```

```
## Warning: NAs introduced by coercion
```

```
energydatats <- energy_data_raw[,1:6]
data<-energydatats[,c(1,5:6)]
nobs<-nrow(data)
#Createvectort-timeindex
t<-1:nobs
#transformingjustthetwocolumnsofinterestintotsobject

tsdata<-ts(data[t,2:3],frequency=12,start=c(1973,1))
```

Stochastic Trend and Stationarity Tests

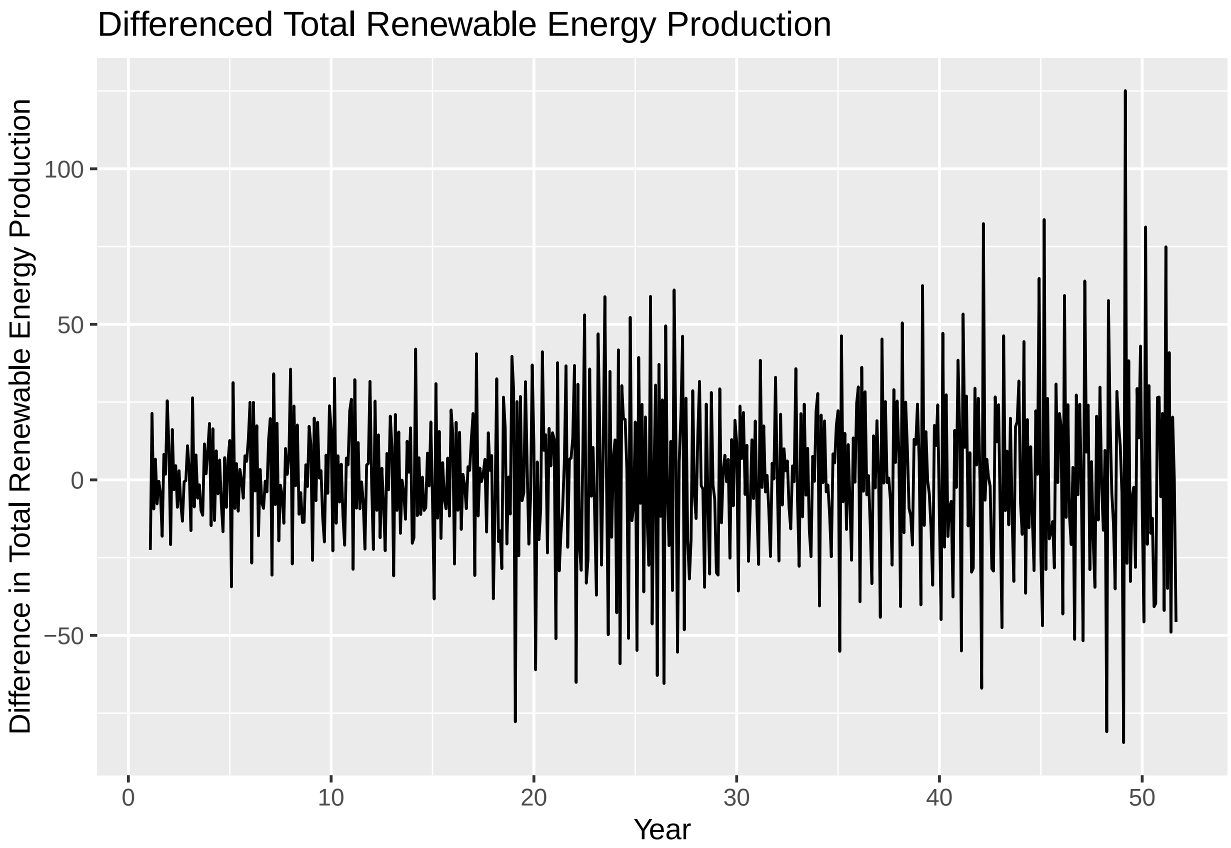
Q1

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?

```
#most current code

TREP_diff1 <- diff(energydata$TREP, lag = 1, differences = 1)

# Convert TREP_diff1 to a time series object
TREP_diff1_ts <- ts(TREP_diff1, start=start(energydata$Month), frequency=12)
```

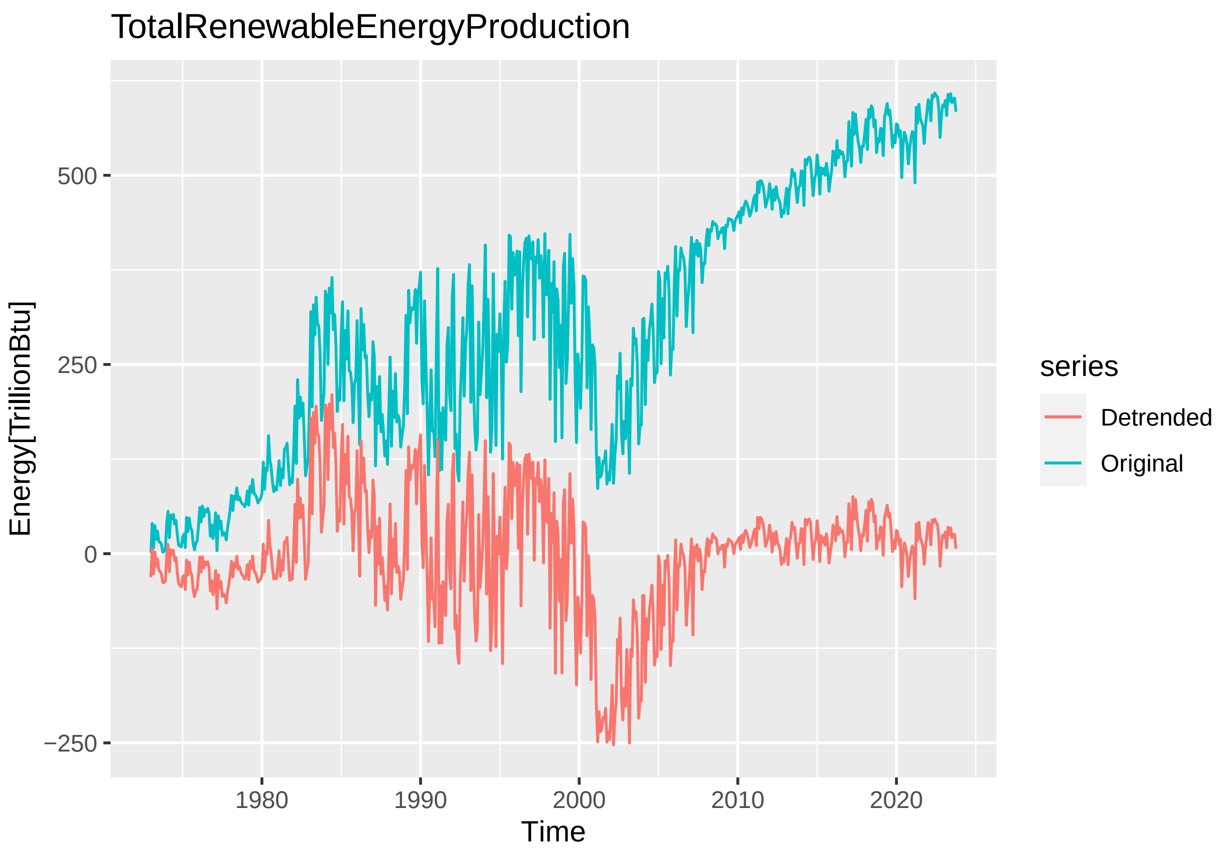
```
# Plot the differenced series
autoplot(TREP_diff1_ts) +
  labs(x = "Year", y="Difference in Total Renewable Energy Production",
       title = "Differenced Total Renewable Energy Production")
```



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 orinal 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.

```
regmodel_renewable=lm(tsdata[,1]~t,cbind(tsdata[,1], t))

beta0_renewable=regmodel_renewable$coefficients[1]

beta1_renewable=regmodel_renewable$coefficients[2]


renewable_detrend<-tsdata[,1]-(beta0_renewable+beta1_renewable*t)

renewable_detrend=ts(renewable_detrend,frequency=12,start=c(1973,1))

#theplotfrompart(d)
autoplot(tsdata[,1],series="Original")+ autolayer(renewable_detrend,series="Detrended") + ylab("Energy[
```



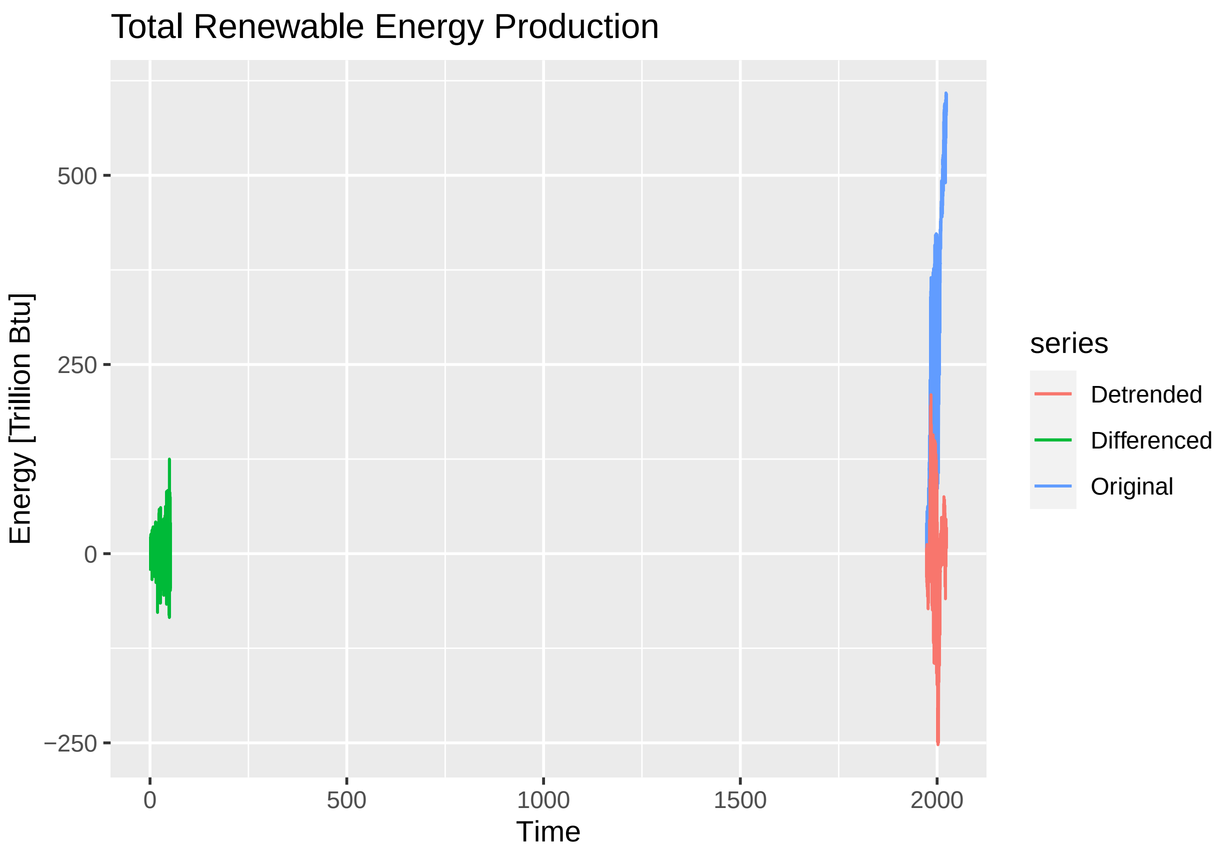
Q3

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.

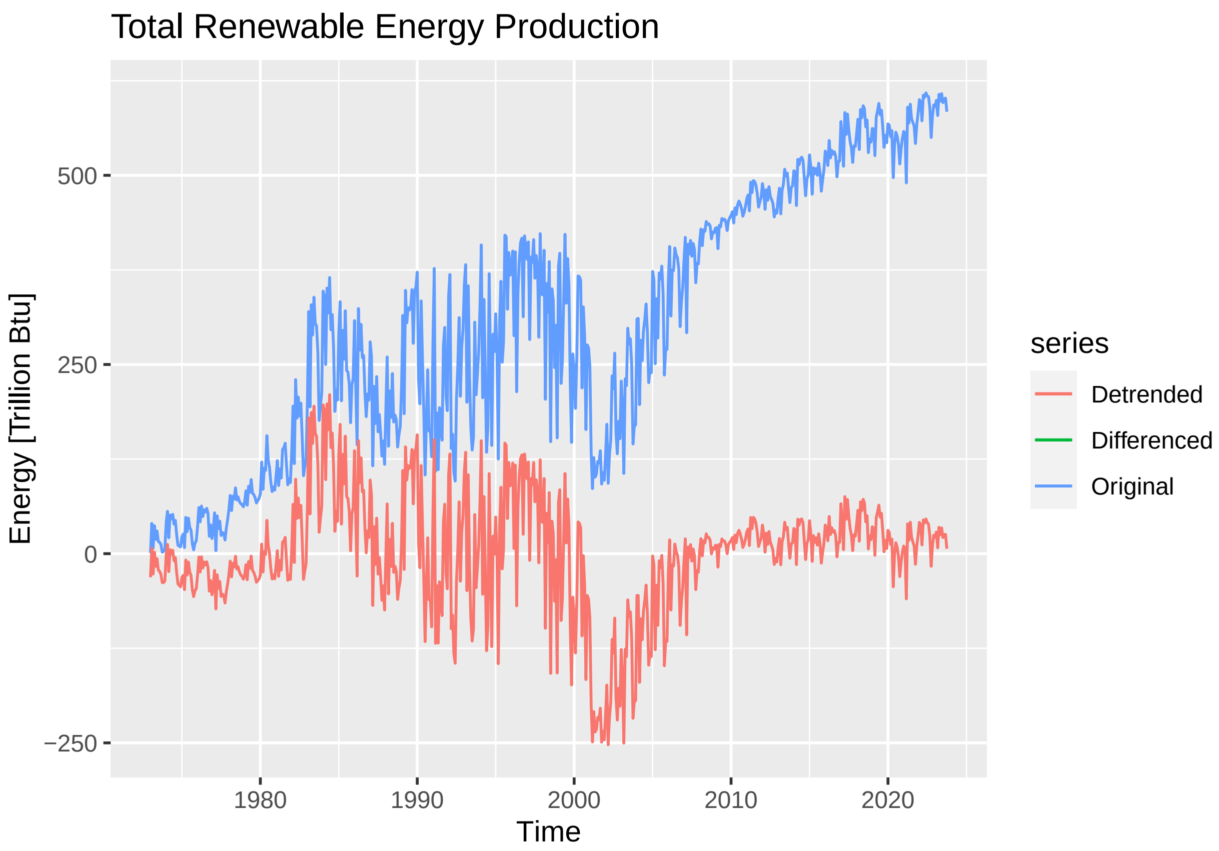
```r
# Plot the original series
autoplot(tsdata[,1], series = "Original") +

# Add the detrended series as a layer
autolayer(renewable_detrend, series = "Detrended") +

# Add the differenced series as another layer
autolayer(TREP_diff1_ts, series = "Differenced") +

# Label the y-axis
ylab("Energy [Trillion Btu]") +

# Add a title
ggtitle("Total Renewable Energy Production")
```

```
## Warning: Removed 1 row containing missing values (‘geom_line()‘).
```



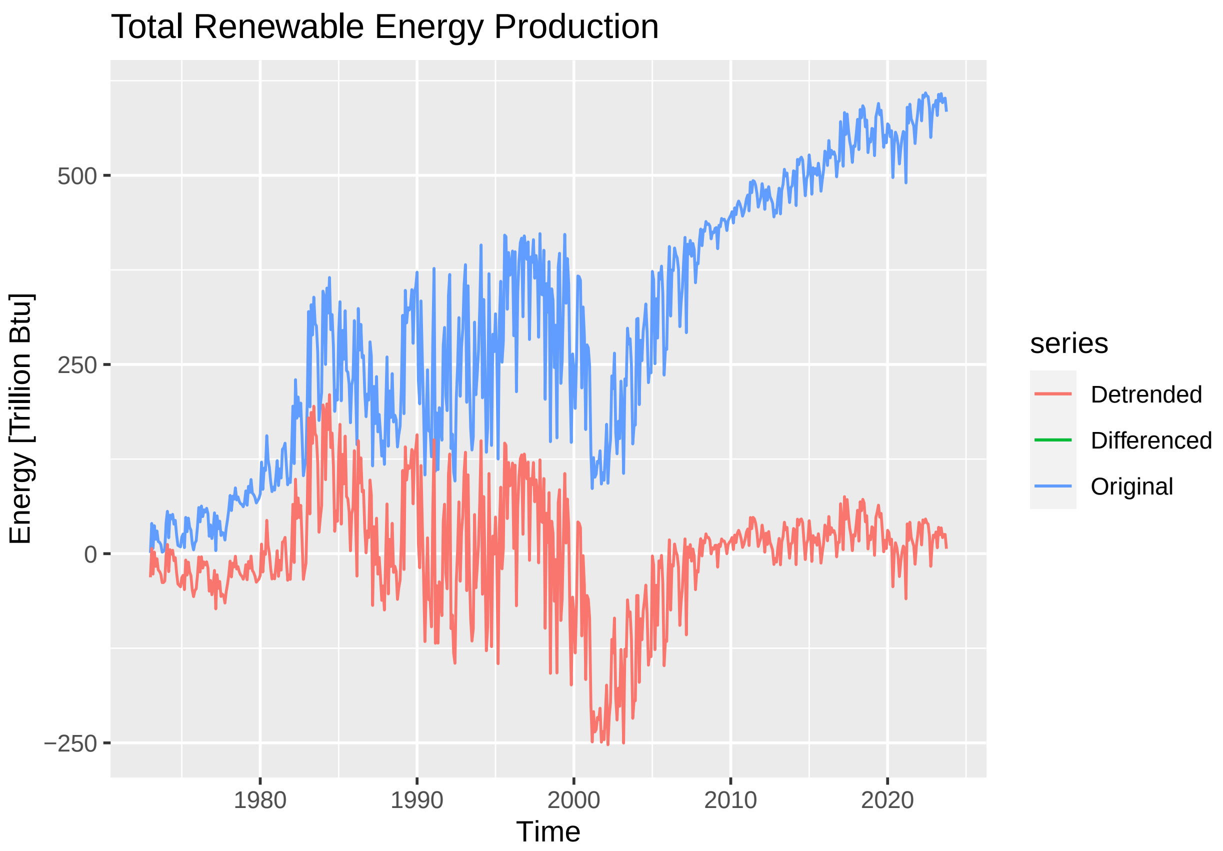
```r
#Determine the x-axis limits
xlim <- range(time(tsdata[,1]))

#Plot the original series with specified x-axis limits
autoplot(tsdata[,1], series="Original") +

#Add the detrended series as a layer
autolayer(renewable_detrend, series="Detrended")+

#Add the differenced series as another layer
autolayer(TREP_diff1_ts, series="Differenced")+

#Set x-axis limits
coord_cartesian(xlim=xlim)+

#Label for the y-axis
ylab("Energy [Trillion Btu]")+

#title
ggtitle("Total Renewable Energy Production")
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```



```
#Remove missing values from the original time series data
tsdata_clean <- na.omit(tsdata[,1])

#This allows us to find out the range of the limts
xlim <-range(time(tsdata_clean))

#Plot the original series
autoplot(tsdata_clean, series="Original")+

#Add the detrended
autolayer(renewable_detrend, series="Detrended")+

#Add the differenced
autolayer(TREP_diff1_ts, series="Differenced")+

#Set x-axis limits for all of the plots
coord_cartesian(xlim=xlim)+
ylab("Energy [Trillion Btu]")+
ggtitle("Total Renewable Energy Production")
```

```
## Warning: Removed 1 row containing missing values (‘geom_line()‘).
```

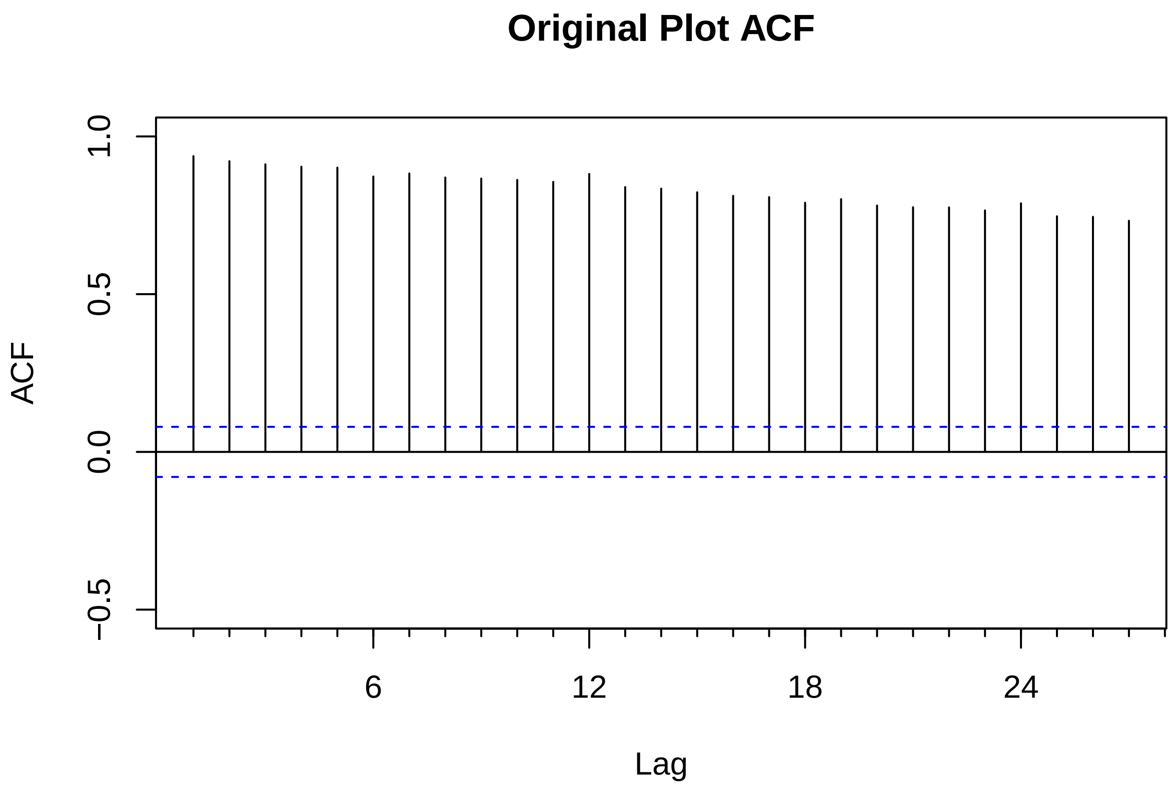


Q4

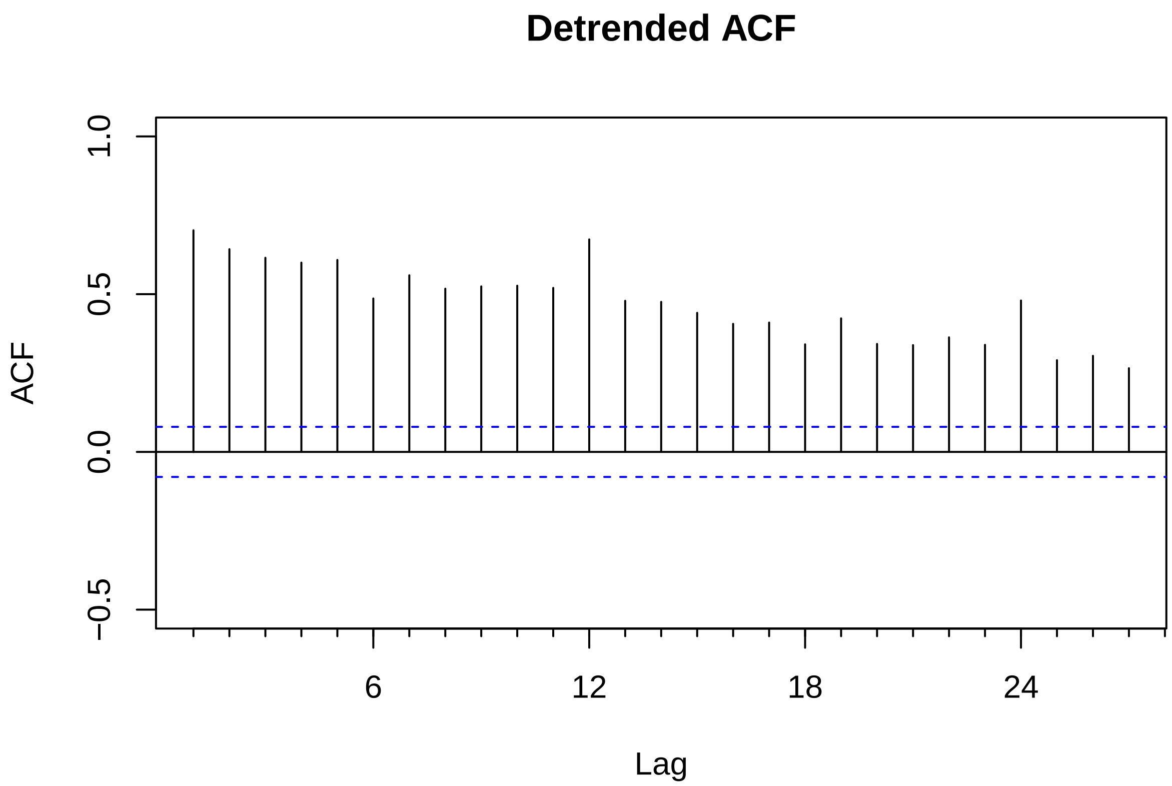
Plot the ACF for the three series and compare the plots. Add the argument `ylim=c(-0.5,1)` to the autoplot() or Acf() function - whichever you are using to generate the plots - to make sure all three y axis have the same limits. Which method do you think was more efficient in eliminating the trend? The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

Most of them still show some type of trend but the best was the detrended ACF and diffrencing model.

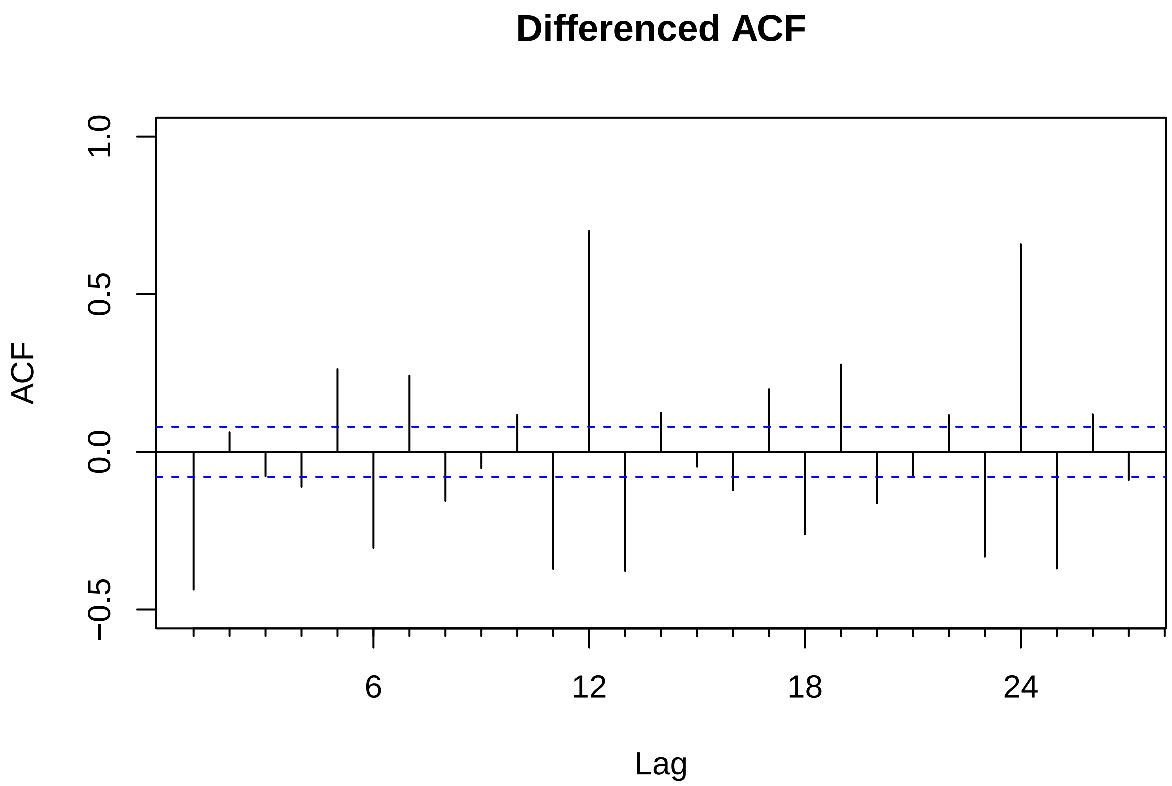
```
#ACF for the original data
Acf(tsdata_clean, ylim=c(-0.5,1), main="Original Plot ACF")
```



```
#Detrended ACF
Acf(renewable_detrend, ylim=c(-0.5,1),main="Detrended ACF")
```



```
#Differenced ACF
Acf(TREP_diff1_ts, ylim=c(-0.5, 1), main="Differenced ACF")
```



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.

From the Man Kendall test we come to learn that our p value is very small and thus confirms that their might be a trend. Also our alternative hypotheses further confirms that their is a trend also since the tau is postive this would be a positive trend. As for ADF, since the pvalue is .1093 we this graph would not be stationary. From this we can obseerve that the trend that we saw in in Q1 is confirmed.

```
#Mann-Kendall test
mk_orginaldata <- mk.test(tsdata[, 1])
print(mk_orginaldata)
```

```
##
##  Mann-Kendall trend test
##
## data:  tsdata[, 1]
## z = 27.746, n = 610, p-value < 2.2e-16
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## 1.395120e+05 2.528196e+07 7.510963e-01
```

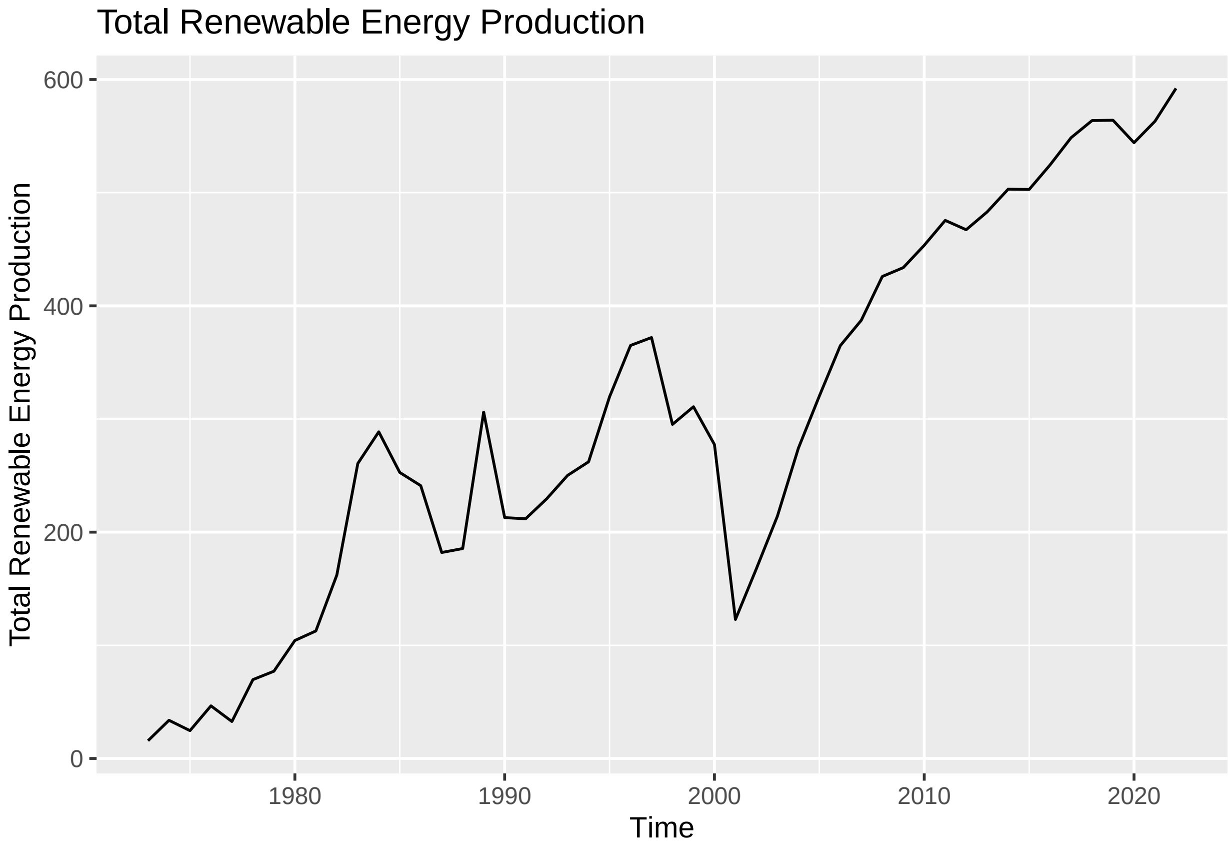
```
#ADF test
adf_orginal <- adf.test(tsdata[, 1])
print(adf_orginal)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  tsdata[, 1]
## Dickey-Fuller = -3.1081, Lag order = 8, p-value = 0.1093
## alternative hypothesis: stationary
```

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().

```
#This is to aggregate the original data series by year
aggregate_data <- aggregate(tsdata[, 1], FUN = mean, by = list(year = as.numeric(format(index(tsdata[, 


autoplot(aggregate_data) +
  ylab("Total Renewable Energy Production") +
  ggtitle("Total Renewable Energy Production")
```



Q7

Apply the Mann Kendal, 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?

The Mann Kendal, Spearman test both give indicative results that the what we find in Q6 is true. Then from the ADF test we have resluts which indicate that series is not stationary.

```r
# Mann-Kendall Test
mk_result <- MannKendall(aggregate_data)
print(mk_result)
```

```
## tau = 0.801, 2-sided pvalue =< 2.22e-16
```

```r
# Spearman Correlation Rank Test
spearman_result <- cor.test(seq_along(aggregate_data), aggregate_data, method = "spearman")
print(spearman_result)
```

```
##
##  Spearman's rank correlation rho
##
## data:  seq_along(aggregate_data) and aggregate_data
## S = 1776, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9147179
```

```
# Augmented Dickey-Fuller (ADF) Test
adf_result <- adf.test(aggregate_data)
print(adf_result)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  aggregate_data
## Dickey-Fuller = -3.0295, Lag order = 3, p-value = 0.1613
## alternative hypothesis: stationary
```