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

Assignment 4 - Due date 02/17/22

Abbhijith Hari Gopal

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.\

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

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.5
```

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

```
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.0.5
```

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.0.5
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.0.5
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.5

##
## Attaching package: 'lubridate'

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

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5

##
## 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
```

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
getwd()
```

```
## [1] "C:/Users/User/Documents/GithubRepos/ENV790_TimeSeriesAnalysis_Sp2022"
```

```
data = read_excel("./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx", skip
```

```
## New names:
## * ‘‘ -> ...1
## * ‘‘ -> ...2
## * ‘‘ -> ...3
## * ‘‘ -> ...4
## * ‘‘ -> ...5
## * ...
```

```r
data_of_interest = data[, 5]
date_of_interest = data[, 1]
head(data_of_interest, 15) #As a check
```

```
## # A tibble: 15 x 1
##      ...5
##     <dbl>
##  1  404.
##  2  361.
##  3  400.
##  4  380.
##  5  392.
##  6  377.
##  7  367.
##  8  354.
##  9  307.
## 10  323.
## 11  338.
## 12  407.
## 13  437.
## 14  400.
## 15  423.
```

```r
head(date_of_interest, 15) #As a check
```

```
## # A tibble: 15 x 1
##    ...1
##    <dttm>
##  1 1973-01-01 00:00:00
##  2 1973-02-01 00:00:00
##  3 1973-03-01 00:00:00
##  4 1973-04-01 00:00:00
##  5 1973-05-01 00:00:00
##  6 1973-06-01 00:00:00
##  7 1973-07-01 00:00:00
##  8 1973-08-01 00:00:00
##  9 1973-09-01 00:00:00
## 10 1973-10-01 00:00:00
## 11 1973-11-01 00:00:00
## 12 1973-12-01 00:00:00
## 13 1974-01-01 00:00:00
## 14 1974-02-01 00:00:00
## 15 1974-03-01 00:00:00
```

```r
colnames(data_of_interest) = c("Total Renewable Energy Production")
colnames(date_of_interest) = c("Date")
data_of_interest$`Total Renewable Energy Production` = as.numeric(data_of_interest$`Total Renewable Ene
class(data_of_interest) #As a check
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```r
new_data = cbind(date_of_interest, data_of_interest)
head(new_data, 10) #As a check
```

```
##          Date 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
```

```r
ncolumns = ncol(new_data)
nmonths = nrow(new_data)

ts_data = ts(new_data[, 2], start = c(1973, 1), frequency = 12)
head(ts_data, 50) #As a check
```

```
##          Jan     Feb     Mar     Apr     May     Jun     Jul     Aug     Sep
## 1973 403.981 360.900 400.161 380.470 392.141 377.232 367.325 353.757 307.006
## 1974 437.467 399.942 423.474 422.323 427.657 409.281 409.719 386.101 353.910
## 1975 392.756 368.278 423.490 405.368 421.283 411.622 398.459 368.230 341.957
## 1976 421.775 396.173 427.044 396.931 415.728 411.555 421.425 398.129 356.984
## 1977 378.521 304.328
##          Oct     Nov     Dec
## 1973 323.453 337.817 406.694
## 1974 343.703 351.633 376.642
## 1975 368.786 383.196 403.696
## 1976 369.186 351.386 360.834
## 1977
```

```r
class(ts_data) #As a check
```

```
## [1] "ts"
```

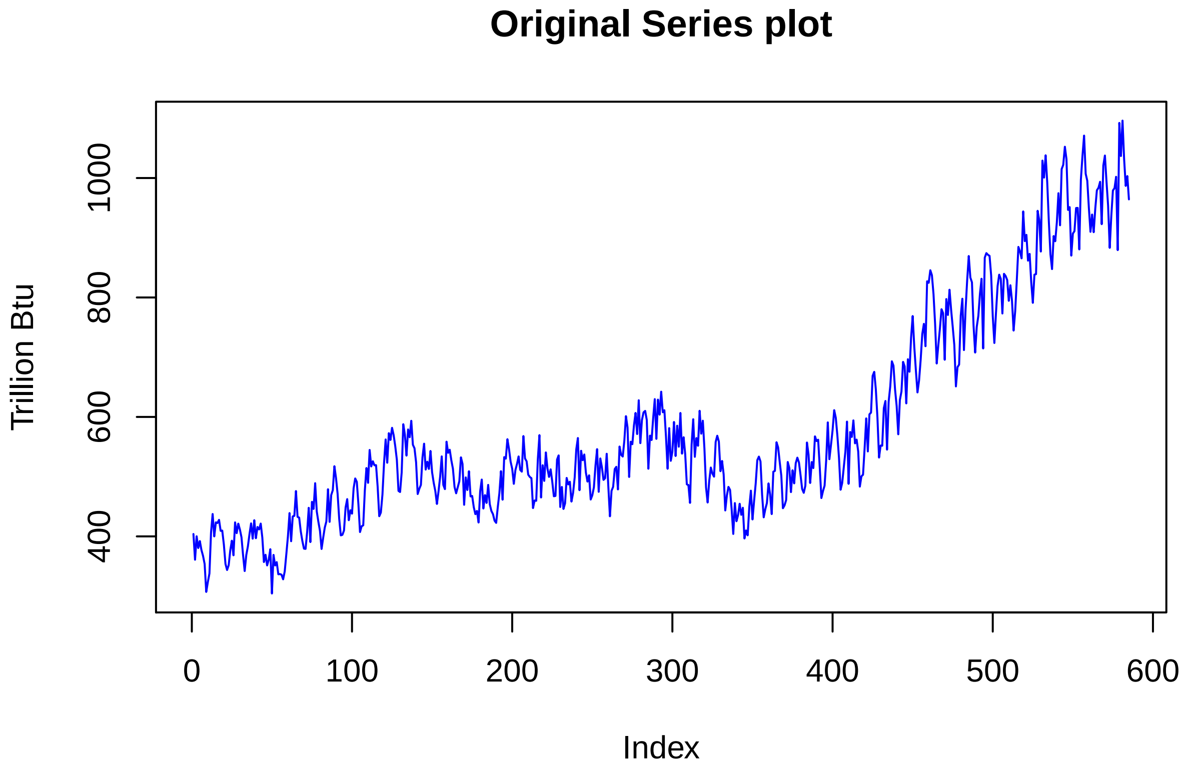
Stochastic Trend and Stationarity Tests

Q1

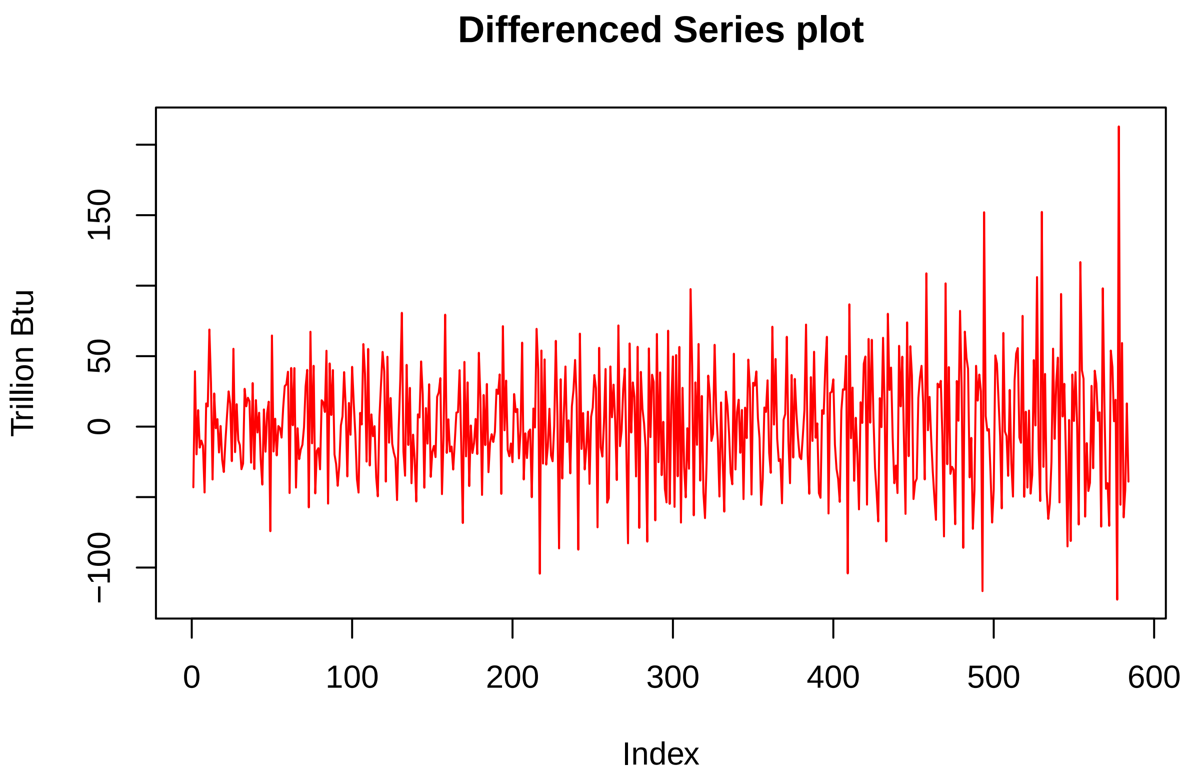
Difference the "Total Renewable Energy Production" series using function diff(). Function diff() is from package base and take three main arguments: * x vector containing values to be differenced; * *lag* integer indicating with lag to use; * *differences* integer indicating how many times series should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

```
plot(new_data[,2], type="l",col="blue",ylab="Trillion Btu", main = "Original Series plot")
```



```
differenced_series = diff(new_data[, 2], lag = 1, differences = 1)
plot(differenced_series, type="l",col="red",ylab="Trillion Btu", main = "Differenced Series plot")
```



```r
head(differenced_series, 10) #As check
```

```
##  [1] -43.081  39.261 -19.691  11.671 -14.909  -9.907 -13.568 -46.751  16.447
## [10]  14.364
```

From the plots, the original series seems to have a positive, increasing trend. The Differenced series, on the other hand, doesn't seem to have any trend. ### 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

```r
t = c(1:nmonths)

renewable_linear_trend = lm(new_data$`Total Renewable Energy Production`~t)
beta0 = as.numeric(renewable_linear_trend$coefficients[1])
beta1 = as.numeric(renewable_linear_trend$coefficients[2])

renewable_detrend = new_data$`Total Renewable Energy Production` - (beta0 + beta1*t)

linear_trend_model=lm(renewable_detrend~t)
linear_trend_beta0 = as.numeric(linear_trend_model$coefficients[1])
linear_trend_beta1 = as.numeric(linear_trend_model$coefficients[2])
summary(linear_trend_model) #detrended data
```

```
## 
## Call:
## lm(formula = renewable_detrend ~ t)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -230.488  -57.869    5.595   62.090  261.349 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.403e-13  8.026e+00       0        1
## t            7.724e-16  2.373e-02       0        1
## 
## Residual standard error: 96.93 on 583 degrees of freedom
## Multiple R-squared:  2.799e-30,  Adjusted R-squared:  -0.001715 
## F-statistic: 1.632e-27 on 1 and 583 DF,  p-value: 1
```

```
print(paste("The slope is", linear_trend_beta1, "and intercept is", linear_trend_beta0))
```

```
## [1] "The slope is 7.72379311179057e-16 and intercept is -3.40291165182084e-13"
```

```
n2 = nrow(new_data) - 1
t1 = c(1:n2)
diff_lin_trend = lm(differenced_series~t1)
beta0 = as.numeric(diff_lin_trend$coefficients[1])
beta1 = as.numeric(diff_lin_trend$coefficients[2])
summary(diff_lin_trend) #differenced data
```

```
## 
## Call:
## lm(formula = differenced_series ~ t1)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -124.741  -28.492   -0.064   28.928  210.971 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.101839   3.438199  -0.030    0.976
## t1           0.003628   0.010184   0.356    0.722
## 
## Residual standard error: 41.49 on 582 degrees of freedom
## Multiple R-squared:  0.000218,   Adjusted R-squared:  -0.0015 
## F-statistic: 0.1269 on 1 and 582 DF,  p-value: 0.7218
```

```
print(paste("The slope is", beta1, "and intercept is", beta0))
```

```
## [1] "The slope is 0.00362791662976466 and intercept is -0.101838559411645"
```

From the output of the regression analysis for detrended and differenced series, we can observe that for detrended series, the slope is $7.7237931 \times 10^{-16}$ and intercept is $-3.4029117 \times 10^{-13}$; for differenced series,

the slope is 0.0036279 and intercept is -0.1018386.
Interpretation: For detrended series by regression: The renewable energy production increases by $7.7237931 \times 10^{-16}$ trillion Btu in 1 year, keeping all other variables constant.
For differenced series: The renewable energy production increases by 0.0036279 trillion Btu in 1 year, keeping all other variables constant.

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.

```
#Data frame - remember to note include January 1973
renewable_detrend_df = data.frame(renewable_detrend)
nrow(renewable_detrend_df) #As check
```

```
## [1] 585
```

```
differenced_series_df = data.frame(differenced_series)
nrow(differenced_series_df) #As check
```

```
## [1] 584
```

```
new_data = data.frame(new_data)
nrow(new_data) #As check
```

```
## [1] 585
```

```
colnames(renewable_detrend_df) = c("Detrended data")
colnames(differenced_series_df) = c("Differenced data")
truncated_new_data = new_data[-1,]
truncated_renewable_detrend = renewable_detrend_df[-1,]
new_df = cbind(truncated_new_data, truncated_renewable_detrend, differenced_series_df) #new data frame
head(new_df, 5) #As check
```

```
##         Date Total.Renewable.Energy.Production truncated_renewable_detrend
## 2 1973-02-01                           360.900                    35.95655
## 3 1973-03-01                           400.161                    74.33705
## 4 1973-04-01                           380.470                    53.76554
## 5 1973-05-01                           392.141                    64.55603
## 6 1973-06-01                           377.232                    48.76653
##   Differenced data
## 2          -43.081
## 3           39.261
## 4          -19.691
## 5           11.671
## 6          -14.909
```

```
class(new_df) #As check
```

```
## [1] "data.frame"
```

```
nrow(new_df) #As check
```

```
## [1] 584
```

Q4

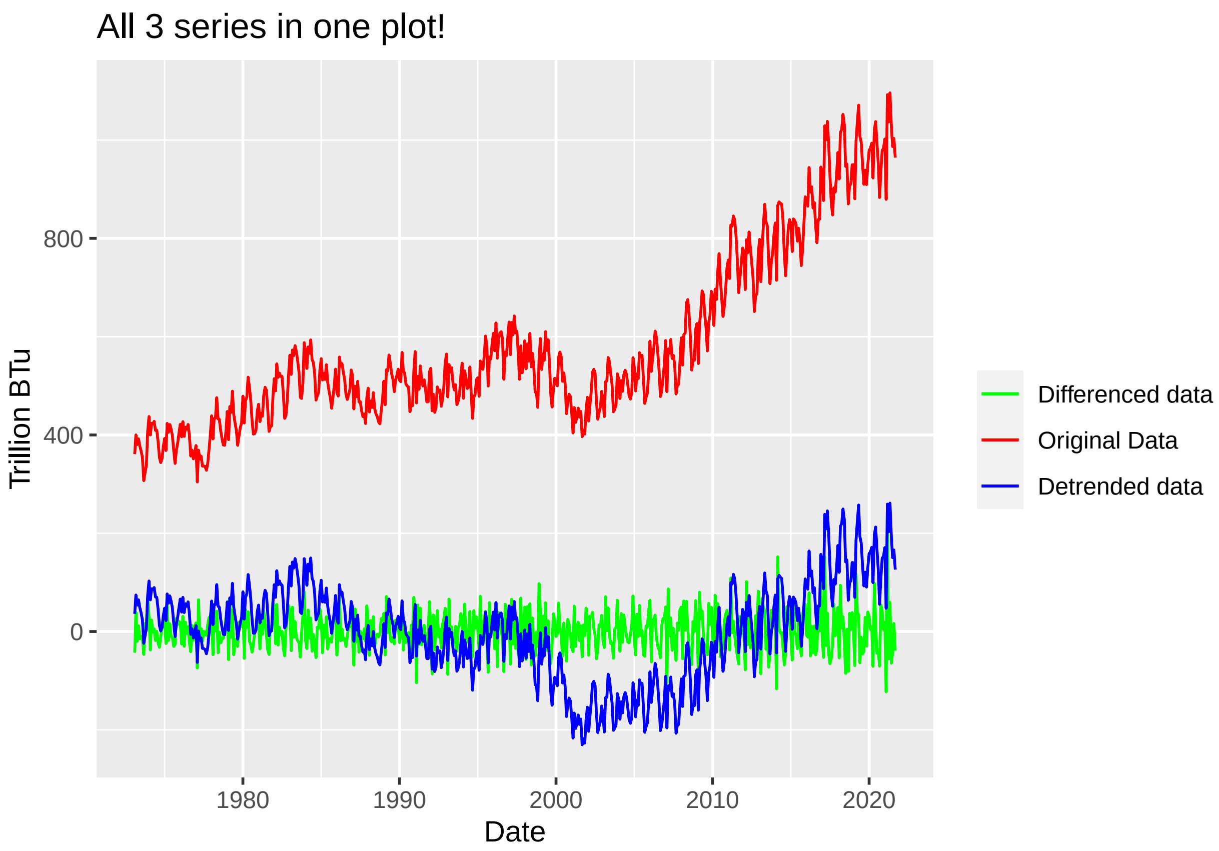
Using ggplot() create a line plot that shows the three series together. Make sure you add a legend to the plot.

```
ggplot(data = new_df) +
  ylab("Trillion BTu") +
  geom_line(aes(Date, new_df$`Differenced data`, colour = "Differenced data")) +
  geom_line(aes(Date, new_df$Total.Renewable.Energy.Production, colour = "Original Data")) +
  geom_line(aes(Date, new_df$truncated_renewable_detrend, colour = "Detrended data")) +
   scale_colour_manual("",
                      breaks = c("Differenced data", "Original Data", "Detrended data"),
                      values = c("Differenced data"="green", "Original Data"="red",
                                 "Detrended data"="blue")) +
  ggtitle("All 3 series in one plot!")
```

```
## Warning: Use of ‘new_df$‘Differenced data‘‘ is discouraged. Use ‘Differenced
## data‘ instead.
```

```
## Warning: Use of ‘new_df$Total.Renewable.Energy.Production‘ is discouraged. Use
## ‘Total.Renewable.Energy.Production‘ instead.
```

```
## Warning: Use of ‘new_df$truncated_renewable_detrend‘ is discouraged. Use
## ‘truncated_renewable_detrend‘ instead.
```



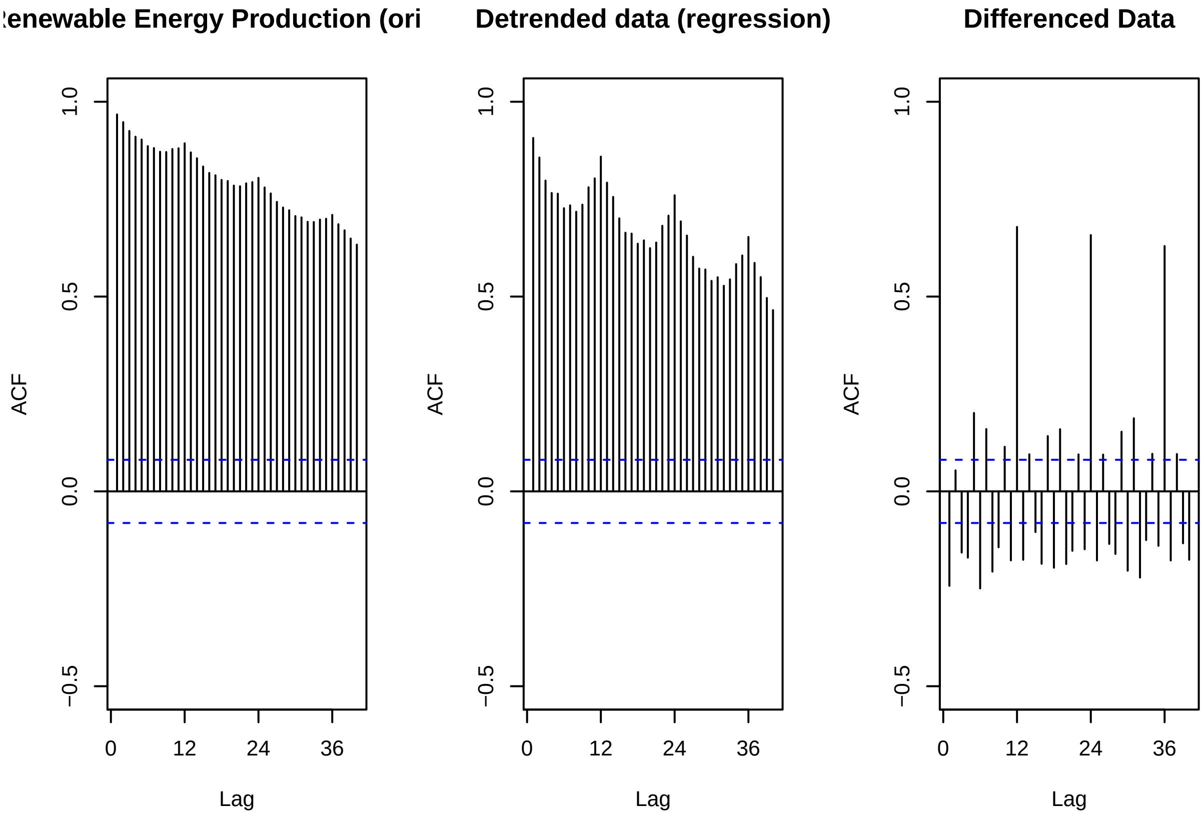
Q5

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?

```
#Compare ACFs
ts_new_object = ts(new_df[, 2:4], frequency = 12)
head(ts_new_object, 5)
```

```
##       Total.Renewable.Energy.Production truncated_renewable_detrend
## Jan 1                           360.900                    35.95655
## Feb 1                           400.161                    74.33705
## Mar 1                           380.470                    53.76554
## Apr 1                           392.141                    64.55603
## May 1                           377.232                    48.76653
##       Differenced data
## Jan 1          -43.081
## Feb 1           39.261
## Mar 1          -19.691
## Apr 1           11.671
## May 1          -14.909
```

```
par(mfrow=c(1,3))
Acf(ts_new_object[, 1], lag.max=40,main=paste("Total Renewable Energy Production (original data)"), ylim
Acf(ts_new_object[, 2], lag.max=40,main=paste("Detrended data (regression)"), ylim=c(-0.5,1))
Acf(ts_new_object[, 3], lag.max=40,main=paste("Differenced Data"), ylim=c(-0.5,1))
```



From the ACF plots, we can observe that differencing seems to remove trend in a more effective way than linear regression. This is because in the detrended ACF plot, detrended by linear regression, it looks like there is still some sort of decreasing trend with increase in lag whereas there isn't any clear trend pattern in the differenced plot

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.

```
SMKtest <- SeasonalMannKendall(ts_data)
print("Results for Seasonal Mann Kendall /n")
```

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

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

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

```
print("Results for ADF test /n")
```

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

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

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

From the summary stats for seasonal Mann Kandall test, we can see p-value $<$ 0.05. Therefore, we can reject the null hypothesis which states that there is no deterministic trend in the data. The idea behind the ADF test is to check for existence of stochastic trend and existence unit root in the data set. P value of 0.81 $>$ 0.05 shows that we can't reject the null hypothesis which states that there exists a unit root. Therefore, we need to use a different procedure to remove the trend. ### 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.

```
data_matrix = matrix(ts_data,byrow=FALSE,nrow=12)
```

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

```
data_yearly = colMeans(data_matrix)
head(data_yearly, 50)
```

```
##  [1] 367.5781 395.1543 390.5934 393.9292 350.7473 417.1201 426.9045 452.3618
##  [9] 451.1407 498.3031 541.3011 536.4885 507.0013 509.2615 468.4839 454.7295
## [17] 519.5548 503.3353 505.6488 485.0463 506.8257 498.9286 546.4422 584.2408
## [25] 584.7342 541.0612 542.9657 508.4722 430.1476 477.5752 495.2055 505.2226
## [33] 518.4010 548.8537 542.5307 599.2957 635.4112 692.8135 775.6386 741.0728
## [41] 786.0602 815.7308 812.8228 871.6138 936.3855 962.6813 966.2615 972.2888
## [49] 854.7981
```

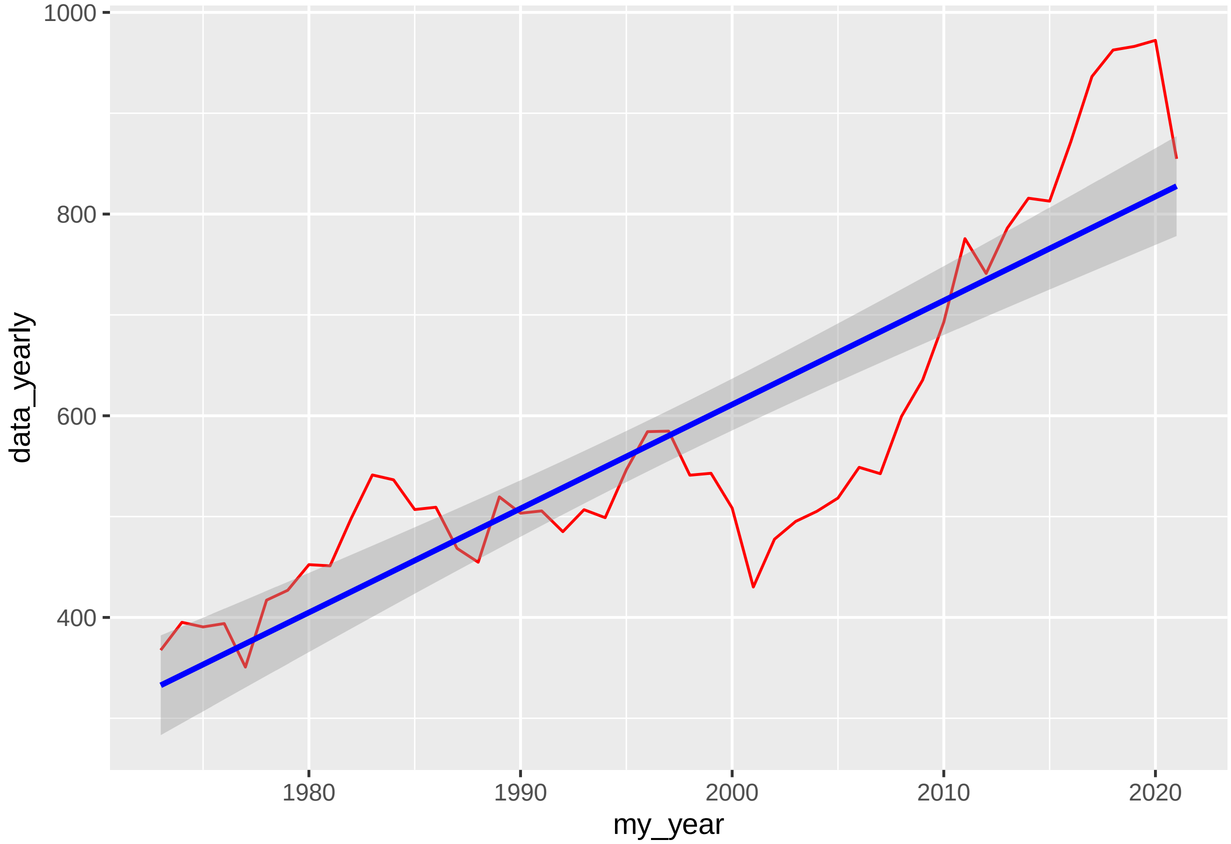
```
my_year = c(year(first(new_data$Date)):year(last(new_data$Date)))
#print(year(last(date_of_interest)))
head(my_year, 50)
```

```
##  [1] 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987
## [16] 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002
## [31] 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017
## [46] 2018 2019 2020 2021
```

```
data_new_yearly = data.frame(my_year, data_yearly)

ggplot(data_new_yearly, aes(x = my_year, y = data_yearly)) +
  geom_line(color = "red") +
  geom_smooth(color = "blue", method = "lm")
```

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



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?

```r
test1 = ts(data_yearly, start = 1973, frequency = 1)
print("Results of Mann Kendall on average yearly series")
```

```
## [1] "Results of Mann Kendall on average yearly series"
```

```r
print(summary(MannKendall(test1)))
```

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

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

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

```r
sp = cor.test(my_year, test1, method="spearman")
print(sp)
```

```
##
##  Spearman's rank correlation rho
##
## data:  my_year and test1
## S = 2578, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.8684694
```

```r
print("Results for ADF test on yearly data/n")
```

```
## [1] "Results for ADF test on yearly data/n"
```

```r
print(adf.test(test1, alternative = "stationary"))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  test1
## Dickey-Fuller = -2.2085, Lag order = 3, p-value = 0.4907
## alternative hypothesis: stationary
```

After aggregating the data yearly, we can see that for the Seasonal Mann Kandall test, the p value, once again is less than 0.05. Therefore, we have the same conclusion we derived in Q6 - there is a deterministic trend. The Spearmann correlation test suggests that the presence of deterministic test too since its p-value is also less than 0.05. Moreover, the positive value of s-score tells us the trend is an increasing trend. The ADF tests' conclusion is also in line with what we got in Q6. The p-value for the aggregated data by year is 0.4907 $>$ 0.05. Therefore, there exists a unit root in the yearly aggregated data too