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

Assignment 4 - Due date 02/12/24

Student Name

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(forecast)
library(tseries)
library(ggplot2)
library(Kendall)
library(readxl)
library(cowplot)
```

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
energy_data <- read_excel(
  path="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
  skip = 12,
  sheet="Monthly Data",
  col_names=FALSE
  )
```

```r
#Now let's extract the column names from row 11 only
read_col_names <- read_excel(
  path="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
  skip = 10,
  n_max = 1,
  sheet="Monthly Data",
  col_names=FALSE
  )

colnames(energy_data) <- read_col_names
nobs <- nrow(energy_data)

#transforming just the two columns of interest into ts object
ts_renewable_data <- ts(as.data.frame(energy_data[,5]),
                        frequency=12,
                        start=c(1973,1)
                        )
#frequency=12 because of monthly data. Data starts from Jan 1973.

head(ts_renewable_data,24)
```

```
##          Jan     Feb     Mar     Apr     May     Jun     Jul     Aug     Sep
## 1973 219.839 197.330 218.686 209.330 215.982 208.249 207.800 203.432 185.300
## 1974 231.010 210.188 226.384 223.218 227.793 218.976 221.909 214.197 200.900
##          Oct     Nov     Dec
## 1973 193.514 195.326 220.755
## 1974 200.312 200.068 211.046
```

Stochastic Trend and Stationarity Tests

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

Q1

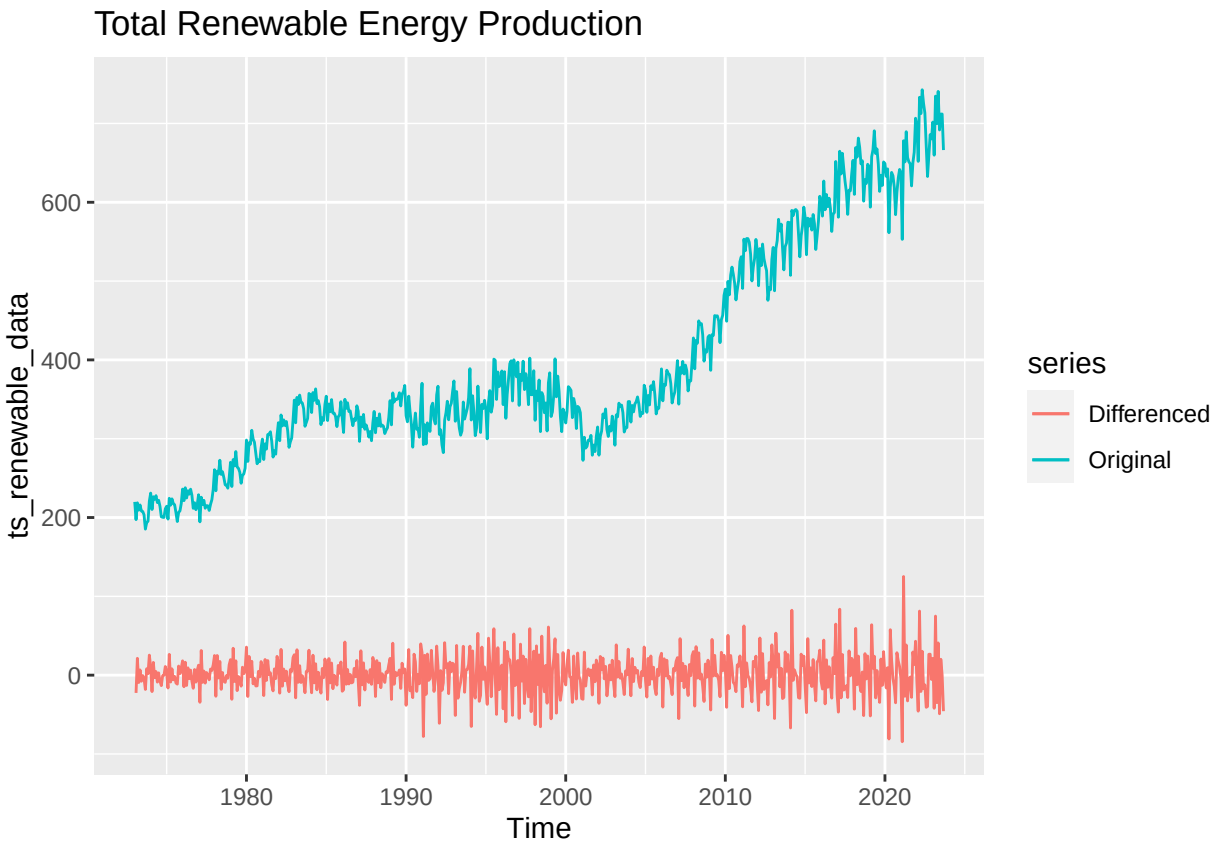
Difference the "Total Renewable Energy Production" series using function diff(). Function diff() is from package base and take three main arguments: * *x* vector containing values to be differenced; * *lag* integer indicating with lag to use; * *differences* integer indicating how many times series should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

```r
diff_renewableP <- diff(ts_renewable_data,lag=1,differences=1)

head(diff_renewableP,24)
```

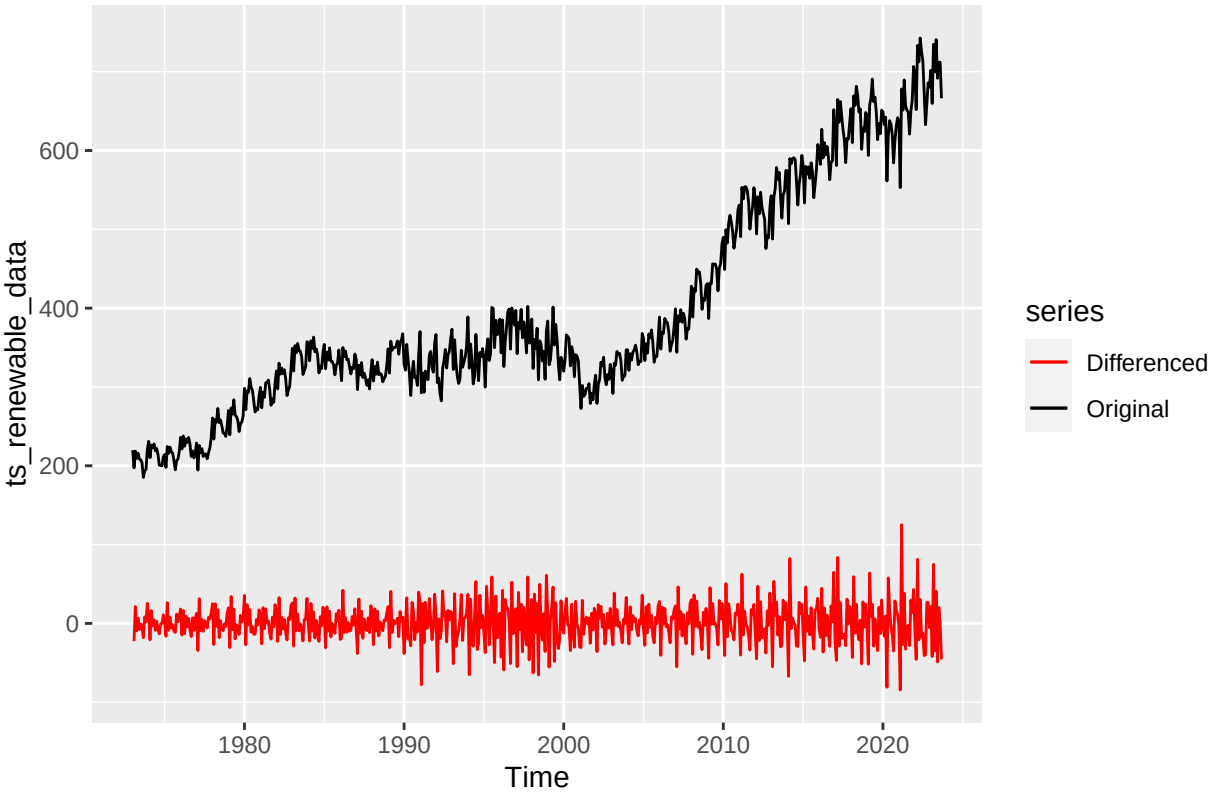
```
##          Jan     Feb     Mar     Apr     May     Jun     Jul     Aug     Sep
## 1973         -22.509  21.356  -9.356   6.652  -7.733  -0.449  -4.368 -18.132
## 1974  10.255 -20.822  16.196  -3.166   4.575  -8.817   2.933  -7.712 -13.297
## 1975   3.273
##          Oct     Nov     Dec
## 1973   8.214   1.812  25.429
## 1974  -0.588  -0.244  10.978
## 1975
```

```
#Note we lost January 1973 observation!

autoplot(ts_renewable_data,series="Original") +
  autolayer(diff_renewableP, series="Differenced")+
  ggtitle("Total Renewable Energy Production")
```



If you want to change defualt colors use teh function `scale_color_manual()` as shown below.

```
autoplot(ts_renewable_data,series="Original") +
  autolayer(diff_renewableP, series="Differenced") +
  scale_colour_manual(
    values=c("Original"="black","Differenced"="red"))
```



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.

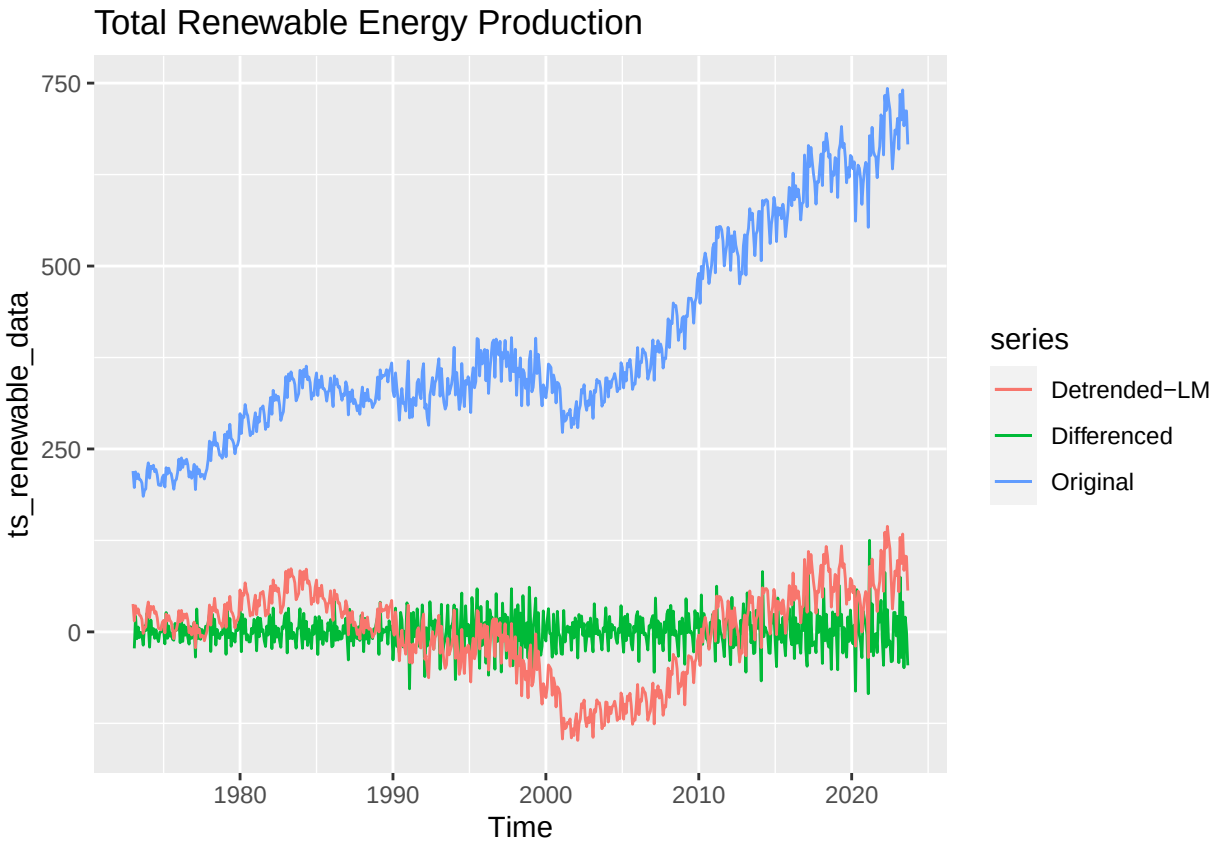
```
t <- 1:nobs

regmodel <- lm(ts_renewable_data ~ t)

#save the regression coefficients for further analysis
beta0 <- regmodel$coefficients[1]
beta1 <- regmodel$coefficients[2]

detrend_renewableP <- ts_renewable_data - (beta0 + beta1*t)
```

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.

```
autoplot(ts_renewable_data,series="Original") +
  autolayer(diff_renewableP, series="Differenced")+
  autolayer(detrend_renewableP,series="Detrended-LM")+
  ggtitle("Total Renewable Energy Production")
```

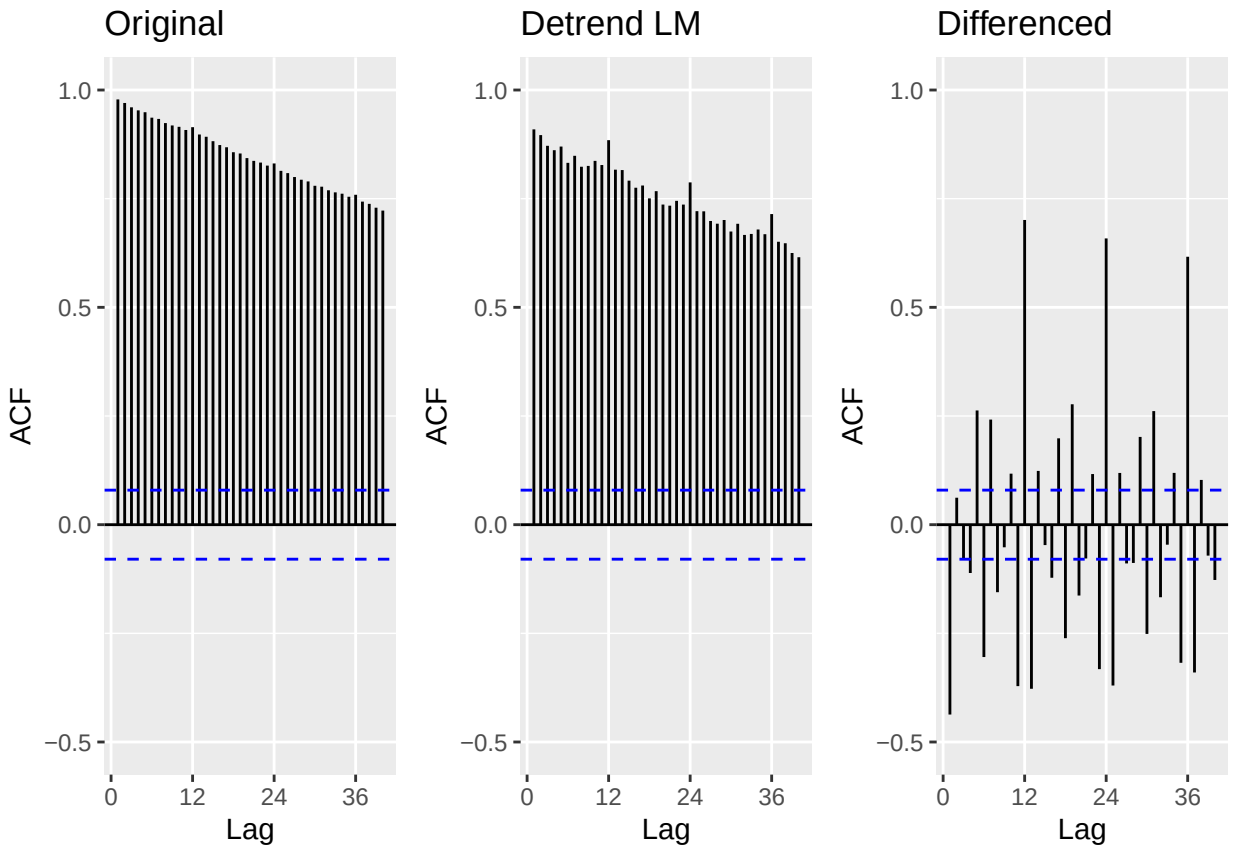


Q4

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 ACFs
plot_grid(
  autoplot(Acf(ts_renewable_data,lag=40,plot=FALSE),ylim=c(-0.5,1),main="Original"),
  autoplot(Acf(detrend_renewableP,lag=40,plot=FALSE),ylim=c(-0.5,1),main="Detrend LM"),
  autoplot(Acf(diff_renewableP,lag=40,plot=FALSE),ylim=c(-0.5,1),main="Differenced"),
  nrow=1
  )
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: ‘ylim‘ and ‘ma
## Ignoring unknown parameters: ‘ylim‘ and ‘main‘
## Ignoring unknown parameters: ‘ylim‘ and ‘main‘
```



Answer: The differencing method was more effective in reducing the magnitude of the correlation coefficients (ACF values). It seems like the differenced series have some seasonality left since the correlation coefficients spieks at the seasonal lags (12,24,36).

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 a different procedure to remove the trend.

```
summary(SeasonalMannKendall(ts_renewable_data))
```

```
## Score =  11865 , Var(Score) = 179299
## denominator =  15149.5
## tau = 0.783, 2-sided pvalue =< 2.22e-16
```

```
#p value < 0.05 then reject null hypothesis, data follow a trend

#If you want to know more about how the trend for each season (i.e. month) use the Seasonal Mann-Kendal
library(trend)
summary(trend::smk.test(ts_renewable_data))
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: ts_renewable_data
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                        S    varS   tau     z   Pr(>|z|)
## Season 1:   S = 0   991 15158.3 0.777 8.041 8.9116e-16 ***
## Season 2:   S = 0   987 15158.3 0.774 8.009 1.1612e-15 ***
## Season 3:   S = 0   973 15158.3 0.763 7.895 2.9081e-15 ***
## Season 4:   S = 0   975 15158.3 0.765 7.911 2.5526e-15 ***
## Season 5:   S = 0   975 15158.3 0.765 7.911 2.5526e-15 ***
## Season 6:   S = 0   991 15158.3 0.777 8.041 8.9116e-16 ***
## Season 7:   S = 0  1019 15158.3 0.799 8.268 < 2.22e-16 ***
## Season 8:   S = 0  1039 15158.3 0.815 8.431 < 2.22e-16 ***
## Season 9:   S = 0  1015 15158.3 0.796 8.236 < 2.22e-16 ***
## Season 10:   S = 0  973 14291.7 0.794 8.131 4.2702e-16 ***
## Season 11:   S = 0  968 14290.7 0.791 8.089 6.0107e-16 ***
## Season 12:   S = 0  959 14291.7 0.783 8.014 1.1146e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
print(adf.test(ts_renewable_data))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_renewable_data
## Dickey-Fuller = -1.24, Lag order = 8, p-value = 0.9
## alternative hypothesis: stationary
```

```
#p value > 0.05 then accept null hypothesis, data has a unit root, i.e., stochastic trend
```

> Answer: Both tests lead to the conclusion that the series has a trend. The SMK shows that all seasonal of the year have an increasing trend. The ADF shows that the series has a unit root, so the existing trend is stochastic. Remember from class that a stochastic trend cannot be effectilively removed with a linear trend model which matches the results from A3 (or Q3). A stochastic trend can only be removed by differencing and it matches what we observed in Q1 when we plotted the differenced series and the trend seems to have been completely removed.

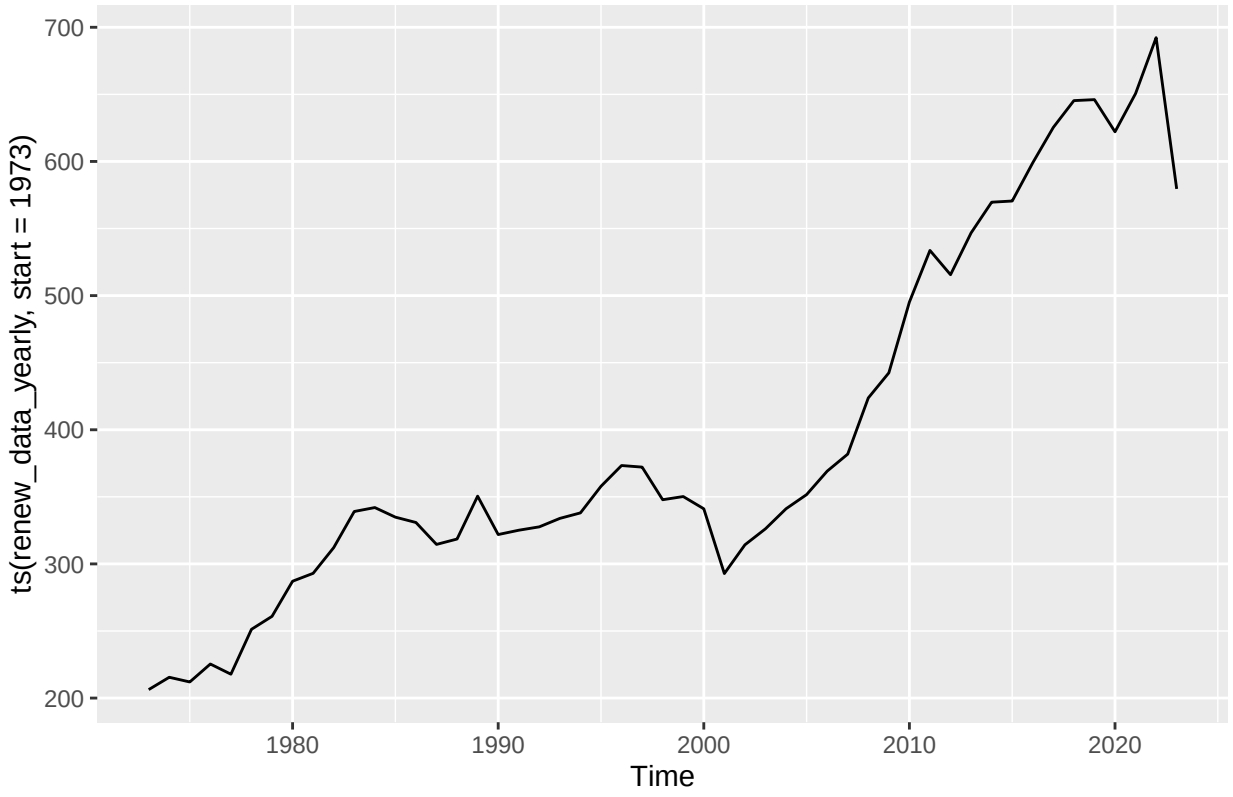
Q6

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().

```r
#Group data in yearly steps instances
renew_data_matrix <- matrix(energy_data$`Total Renewable Energy Production`,byrow=FALSE,nrow=12)
```

```
## Warning in matrix(energy_data$`Total Renewable Energy Production`, byrow =
## FALSE, : data length [609] is not a sub-multiple or multiple of the number of
## rows [12]
```

```r
renew_data_yearly <- colMeans(renew_data_matrix)

autoplot(ts(renew_data_yearly,start = 1973))
```



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?

```r
summary(MannKendall(ts(renew_data_yearly)))
```

```
## Score =  1019 , Var(Score) = 15158.33
## denominator =  1275
## tau = 0.799, 2-sided pvalue =< 2.22e-16
```

```r
#p value < 0.05 then reject null hypothesis, data follow a trend

t <- 1:length(renew_data_yearly)
print(cor.test(x=ts(renew_data_yearly),y=t,method="spearman"))
```

```
##
##  Spearman's rank correlation rho
##
## data:  ts(renew_data_yearly) and t
## S = 1908, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9136652
```

```
#similar results, data follow a trend

print(adf.test(ts(renew_data_yearly)))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts(renew_data_yearly)
## Dickey-Fuller = -2.0953, Lag order = 3, p-value = 0.5361
## alternative hypothesis: stationary
```

```
#p value > 0.05 then accept null hypothesis, data has a unit root, i.e., stochastic trend
```

Answer: The test results on the agregated yearly series matches the results from Q6, the three test lead to the conclusion that series has an increasing trend, but according to ADF it is a stochastic trend.