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

Assignment 4 - Due date 02/12/24

Yilun Zhu

## 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(stats)
library(cowplot)
```

```
##
## Attaching package: 'cowplot'

## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
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
```

```
knitr::opts_chunk$set(echo = TRUE,tidy.opts=list(width.cutoff=80), tidy= FALSE)
```

## 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
Renew_Eng <- read.csv(file="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.csv
#Take one column
TREP_Eng <- Renew_Eng[,5]
#Transfer Ren_Date from a factor to date, in order to use in plot
Ren_Date <- ym(Renew_Eng[,1])
#Use cbind to only bind two rows, it will change date to numeric
#so here use cbind.data.frame
TREP_data <- cbind.data.frame(Ren_Date, TREP_Eng)
nobs <- nrow(TREP_data)
ts_TREP_Eng <- ts(TREP_data[,2], start = c(1973,1), frequency = 12)
```

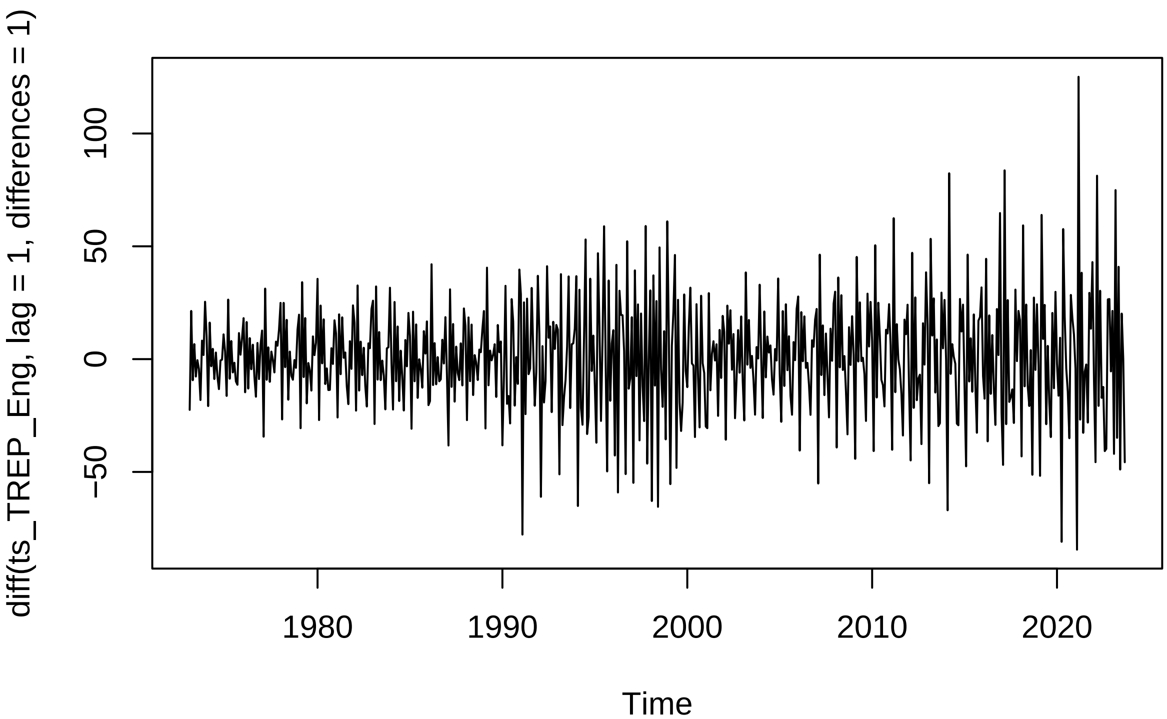
## Stochastic Trend and Stationarity Tests

### 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
plot(diff(ts_TREP_Eng, lag=1, differences = 1))
```




```r
DiffTREP <- diff(ts_TREP_Eng, lag=1, differences = 1)
#Seems that there is almost no trend.
#Or maybe it has a trend of magnitude becoming larger.
```

**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.

```r
t <- 1:nobs
linear_Trend_TREP <- lm(TREP_data[,2]~t)
summary(linear_Trend_TREP)
```

```
##
## Call:
## lm(formula = TREP_data[, 2] ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -148.27  -35.63   11.58   41.51  144.27
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 180.98940    4.90151   36.92   <2e-16 ***
## t             0.70404    0.01392   50.57   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.41 on 607 degrees of freedom
## Multiple R-squared:  0.8081, Adjusted R-squared:  0.8078
## F-statistic:  2557 on 1 and 607 DF,  p-value: < 2.2e-16
```

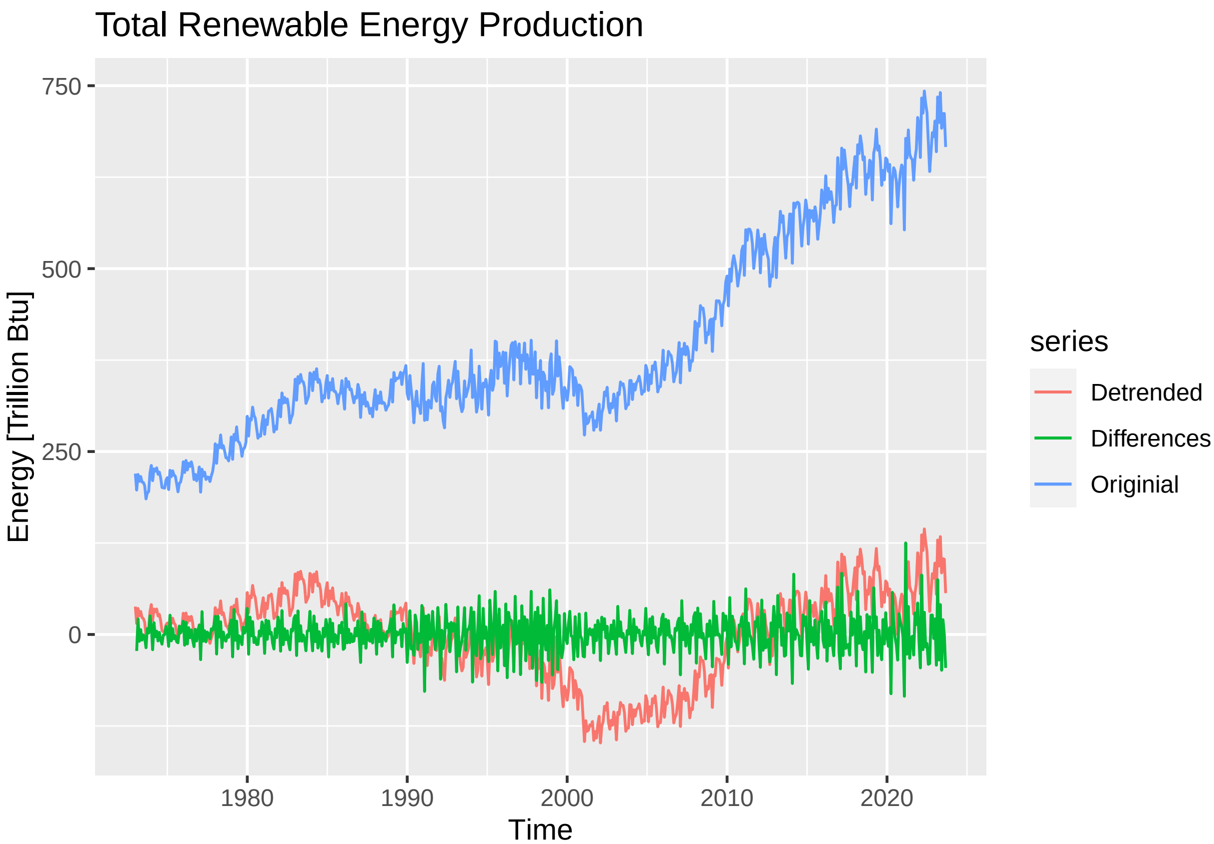
```
TREP_beta0 <- linear_Trend_TREP$coefficients[1]
TREP_beta1 <- linear_Trend_TREP$coefficients[2]

TREP_detrend <- TREP_data[,2] - (TREP_beta0 + TREP_beta1*t)
df_TREPdetrend <- data.frame("date"= TREP_data$Ren_Date, "TREPobserved" = TREP_data[,2], "TREPdetrend" =
```

### 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.

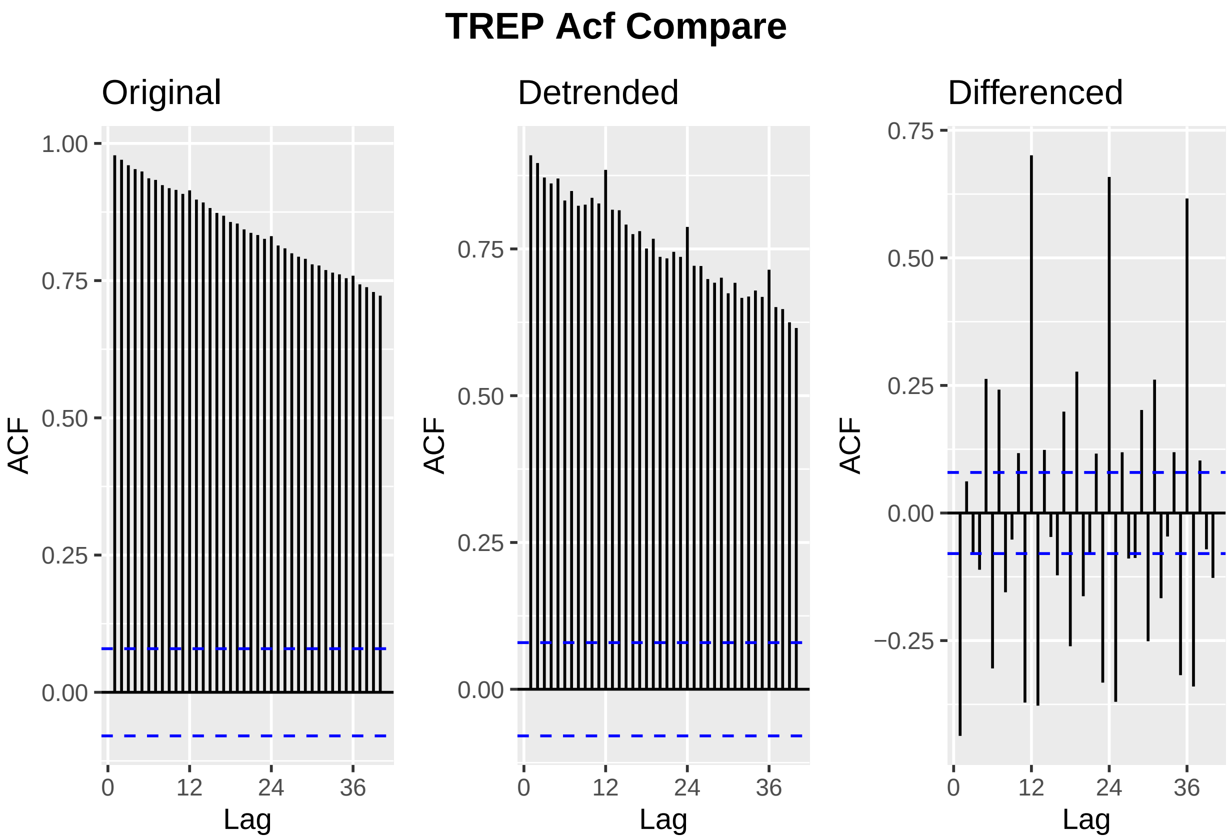
```
#Before autolayer, change TREP_detrend to ts object
ts_TREP_Detrend <- ts(TREP_detrend,start = c(1973,1), frequency = 12)
autoplot(ts_TREP_Eng, series = "Originial")+
  autolayer(ts_TREP_Detrend, series = "Detrended")+
  autolayer(DiffTREP, series = "Differences")+
  ylab("Energy [Trillion Btu]")+
  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?

```
TREP_acf <- autoplot(Acf(ts_TREP_Eng, ylim=c(-0.5,1), lag.max=40, plot = FALSE ))+ ggtitle("Original")
TREP_det_acf <- autoplot(Acf(ts_TREP_Detrend, ylim=c(-0.5,1), lag.max=40, plot = FALSE ))+ ggtitle("Det
TREP_diff_acf <- autoplot(Acf(DiffTREP, ylim=c(-0.5,1), lag.max=40, plot = FALSE ))+ ggtitle("Differenc

plotdraft <- plot_grid(TREP_acf,TREP_det_acf,TREP_diff_acf, nrow = 1, ncol =3)
titledraft <- ggdraw() + draw_label("TREP Acf Compare", fontface='bold')

plot_grid(titledraft,plotdraft,nrow=2,ncol=1,rel_heights = c(0.1,1))
```




```
#Differencing shows great elimination on trend of TREP.
```

### 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.

```
SMKtest <- SeasonalMannKendall(ts_TREP_Eng)
print("Result for seasonal Mann Kendall")
```

```
## [1] "Result for seasonal Mann Kendall"
```

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

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

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

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

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

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

```r
#The SMK test shows TREP is increasing in overall view,
#and since the p-value is extremely small,
#we can reject the hypothesis that TREP is stationary and
#confirm that TREP follows a trend.
#The ADF test shows a great p-value which is 0.9,
#indicating that TREP may has a unit root,
#so we cannot reject hypothesis.
#Based on SMK and ADF test, we can get a conclusion
#that TREP may follow a stochastic trend.
#This outcome doesn't match with the Q2 detrended acf
#but more likely match with differenced TREP.
```

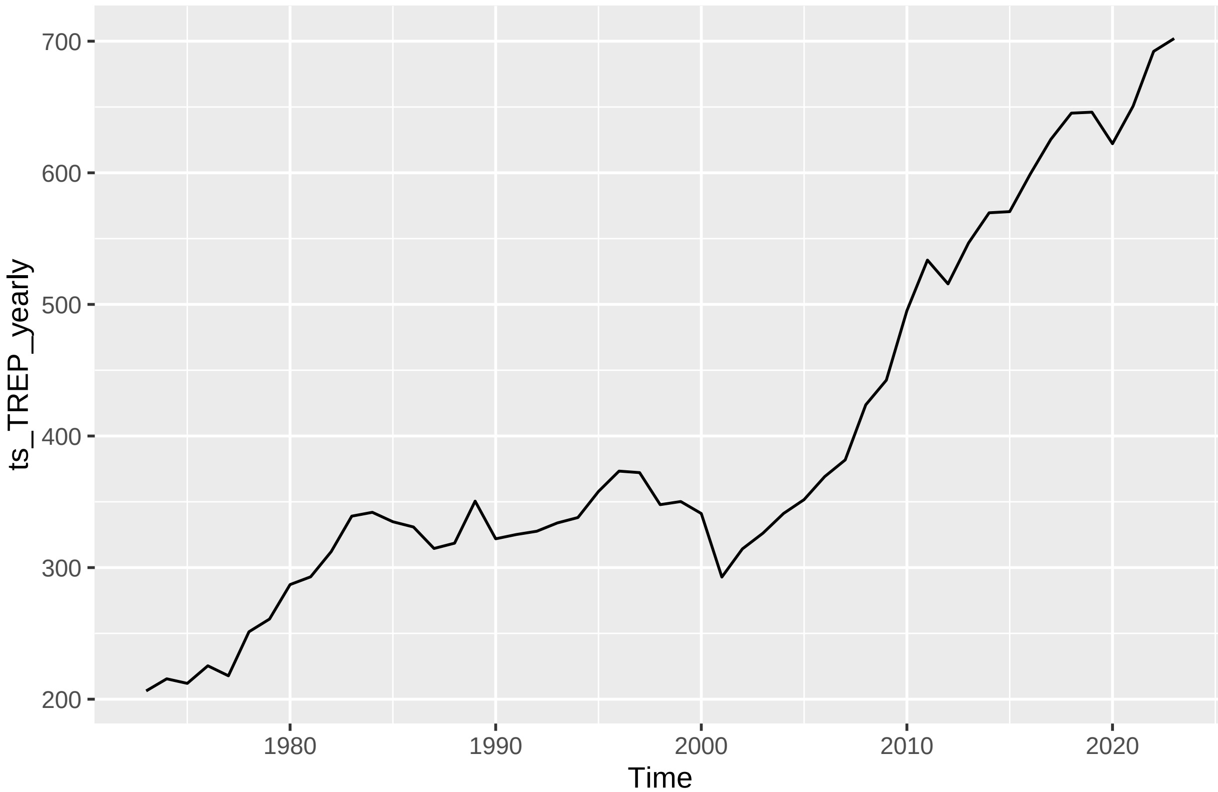
### Q6

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
TREP_matrix <- matrix(ts_TREP_Eng,byrow=FALSE, nrow=12)
```

```
## Warning in matrix(ts_TREP_Eng, byrow = FALSE, nrow = 12): data length [609] is
## not a sub-multiple or multiple of the number of rows [12]
```

```r
TREP_yearly <- colMeans(TREP_matrix)
#Since TREP only has 609 objects, so last row of 2023 lack 3 months data
#and is filled by the first 3 months
#So (mean2023*12-3month)/9 is the actually mean of 2023
delete3 <- sum(TREP_data[1:3,2])
Mean2023 <- ((TREP_yearly[51]*12)-delete3)/9
TREP_yearly <- c(TREP_yearly[1:50],Mean2023)
#Create dataframe of yearly TREP for testing
my_year <- c(year(first(Ren_Date)):year(last(Ren_Date)))
df_TREP_yearly <- data.frame(my_year, TREP_yearly)
ts_TREP_yearly <- ts(df_TREP_yearly[,2], start = c(1973), frequency = 1)

autoplot(ts_TREP_yearly)
```




**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
#MKTest
MKtest <- MannKendall(ts_TREP_yearly)
print("Result for seasonal Mann Kendall")
```

```
## [1] "Result for seasonal Mann Kendall"
```

```r
print(summary(MKtest))
```

```
## Score =  1033 , Var(Score) = 15158.33
## denominator =  1275
## tau = 0.81, 2-sided pvalue =< 2.22e-16
## NULL
```

```r
#Spearman test
TREP_sp_rho=cor.test(ts_TREP_yearly,my_year,method="spearman")
print(TREP_sp_rho)
```

```
##
##  Spearman's rank correlation rho
##
```

```
## data:  ts_TREP_yearly and my_year
## S = 1852, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9161991
```

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

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

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_TREP_yearly
## Dickey-Fuller = -1.0857, Lag order = 3, p-value = 0.9163
## alternative hypothesis: stationary
```

```r
#The result of yearly TREP is in agreement with monthly TREP.
```