ENV 797 - Time Series Analysis for Energy and Environment Applications | Spring 2025

Assignment 4 - Due date 02/11/25

Jingze Dai

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_Sp25.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
list_of_lib <- c(
  "forecast","tseries","Kendall","dplyr","openxlsx","ggplot2", "cowplot")
for (i in list_of_lib){
  library(i,  character.only = TRUE)
}
```

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 - you may copy your code from A3
energy_data <- read.xlsx(
  xlsxFile="../Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
  sheet = "Monthly Data", startRow = 13, colNames = FALSE)

read_col_names  <- read.xlsx(
  xlsxFile="../Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
```

```
  sheet = "Monthly Data", rows = 11, colNames = FALSE)

colnames(energy_data) <- read_col_names
energy_data$Data <- convertToDate(energy_data$Month)

# selecting columns
power_data <- energy_data %>%
  select("Total Renewable Energy Production")

# transforming to time series
ts_power_data <- ts(power_data, start=c(1973,1), frequency=12)
```

Stochastic Trend and Stationarity Tests

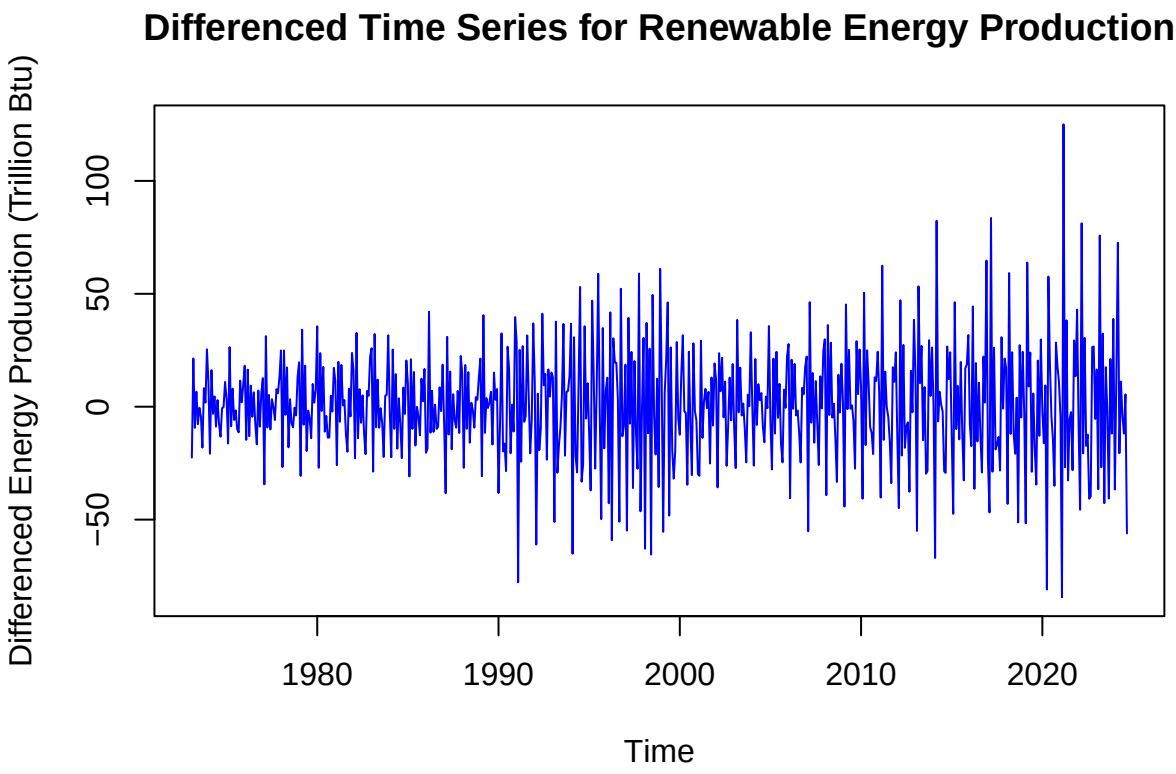
For this part you will work only with the column Total Renewable Energy Production.

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?

```
power_data_diff <- diff(ts_power_data, lag = 1, differences = 1)
plot(power_data_diff,
     type="l",col="blue",
     ylab="Differenced Energy Production (Trillion Btu)",
     main="Differenced Time Series for Renewable Energy Production")
```



Answer: No, the series do not seem to have a trend after differencing it. The original upward trend disappeared.

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, otherwise the code will not work.

```
# fitting linear regression
nobs <- nrow(power_data)
t <- c(1:nobs)
renewable_linear_trend <- lm(power_data[,1] ~ t)
renewable_intercept <- as.numeric(renewable_linear_trend$coefficients[1])
renewable_gradient <- as.numeric(renewable_linear_trend$coefficients[2])

# detrending
renewable_linear_trend <- renewable_intercept  + renewable_gradient * t
ts_renewable_linear <- ts(renewable_linear_trend,start=c(1973,1), frequency=12)

detrend_renewable <- power_data - renewable_linear_trend
ts_detrend_renewable <- ts(detrend_renewable, start = c(1973,1),frequency=12)
```

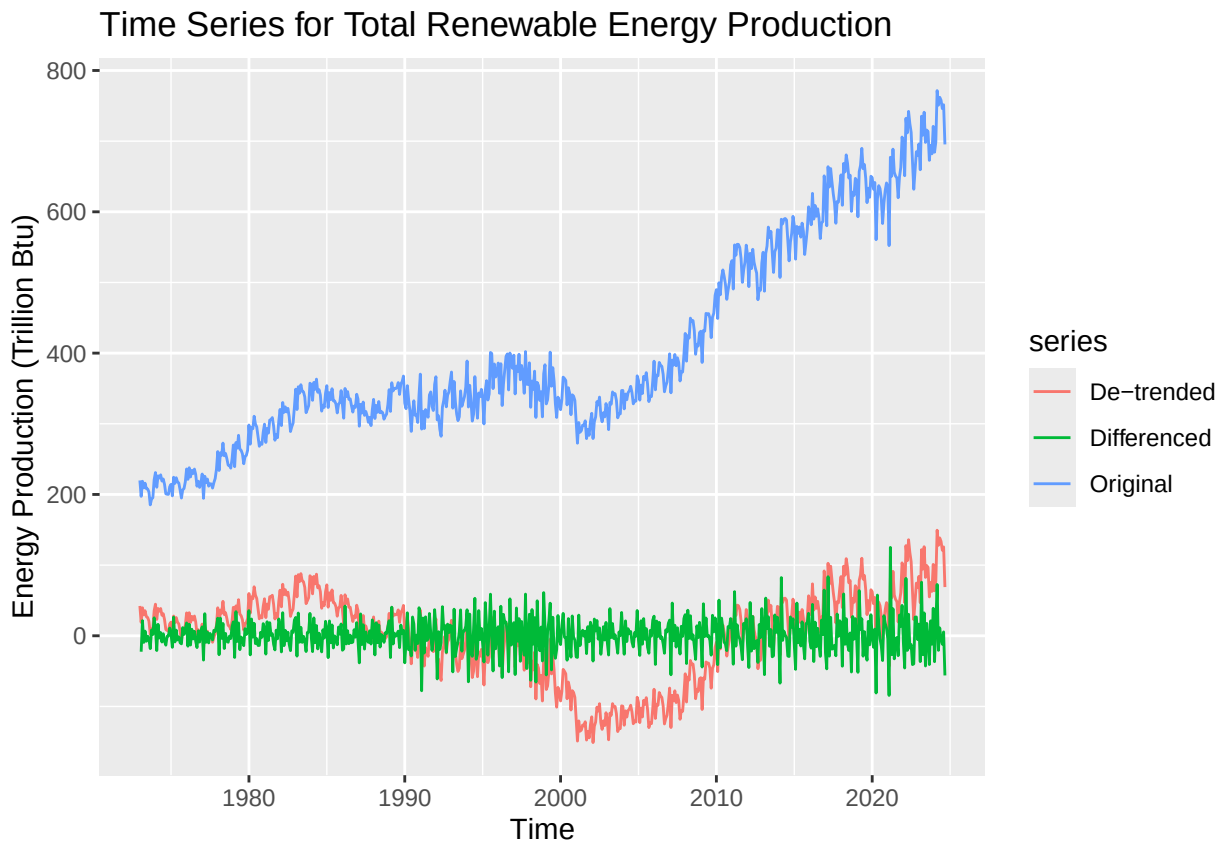
Q3

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 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_power_data, series = "Original") +
  autolayer(ts_detrend_renewable, series = "De-trended")+
  autolayer(power_data_diff, series = "Differenced") +
  ylab("Energy Production (Trillion Btu)") +
  ggtitle("Time Series for Total Renewable Energy Production")
```



Answer: The de-trended time series still shows a certain trend, more specifically, there is a downward trend from 1985 to 2000 and an upward trend from 2000 to 2024. However, the differenced time series is mainly flat and shows no apparent trend. Thus, the differencing method seems to be more efficient in removing trend in this case.

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?

```
acf_orig <- autoplot(Acf(ts_power_data, lag.max=40, plot=FALSE),
                     main="Original", ylim=c(-0.5,1))
```

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

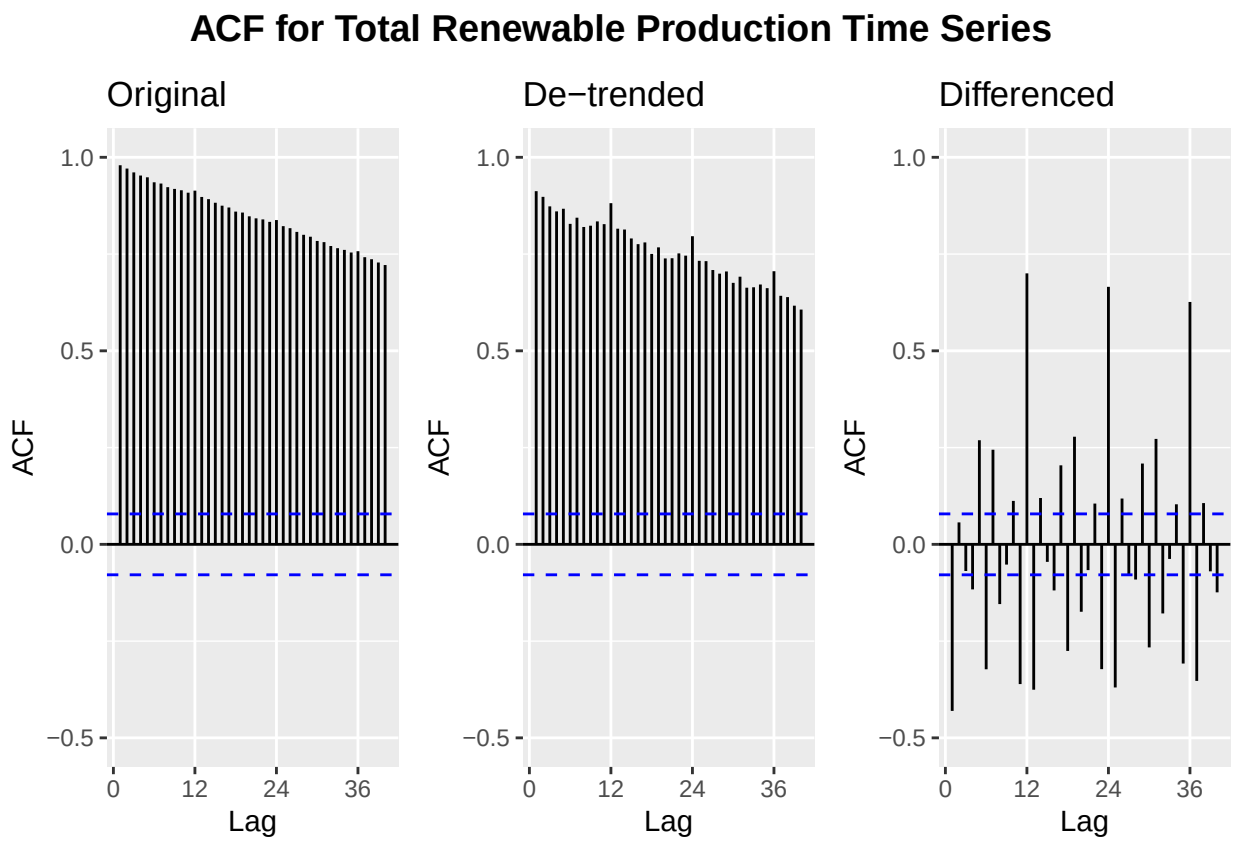
```
acf_detrend <- autoplot(Acf(ts_detrend_renewable, lag.max=40, plot=FALSE),
                        main="De-trended", ylim=c(-0.5,1))
```

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

```
acf_diff <- autoplot(Acf(power_data_diff, lag.max=40, plot=FALSE),
                     main="Differenced",ylim=c(-0.5,1))
```

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

```
plot_row <- plot_grid(acf_orig, acf_detrend, acf_diff, nrow=1, ncol=3)
acf_title <- ggdraw() + draw_label(
  "ACF for Total Renewable Production Time Series", fontface="bold")
plot_grid(acf_title, plot_row, nrow=2, ncol=1, rel_heights=c(0.1,1))
```



Answer: Differencing is more efficient in removing the trend for this time series, because from the ACF plots, the de-trended series only removed a small magnitude of the autocorrelation while the differencing method completely removes the autocorrelation in between those seasonality-induced significant ACF values.

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.

```
print("Results from Seasonal Mann-Kendall Test")
```

```
## [1] "Results from Seasonal Mann-Kendall Test"
```

```
summary(SeasonalMannKendall(ts_power_data))
```

```
## Score =  12468 , Var(Score) = 190008
## denominator =  15758.5
## tau = 0.791, 2-sided pvalue =< 2.22e-16
```

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

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

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_power_data
## Dickey-Fuller = -1.0898, Lag order = 8, p-value = 0.9242
## alternative hypothesis: stationary
```

Answer: The p-value for the seasonal Mann-Kendall test is less than 0.05, indicating a significant deterministic trend. Moreover, the score for the test is 12468, indicating a significant increasing trend. The p-value for the Augmented Dickey-Fuller test is 0.9242, significantly rejecting the alternative hypothesis that the time series is stationary, which means that there is stochastic trend. Moreover, the absolute value of the test score is 1.0898, close to 1, indicating the presence of a unit root, confirming the stochastic trend. The results do match with the plots in previous questions and differencing indeed removed the stochastic trend.

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().

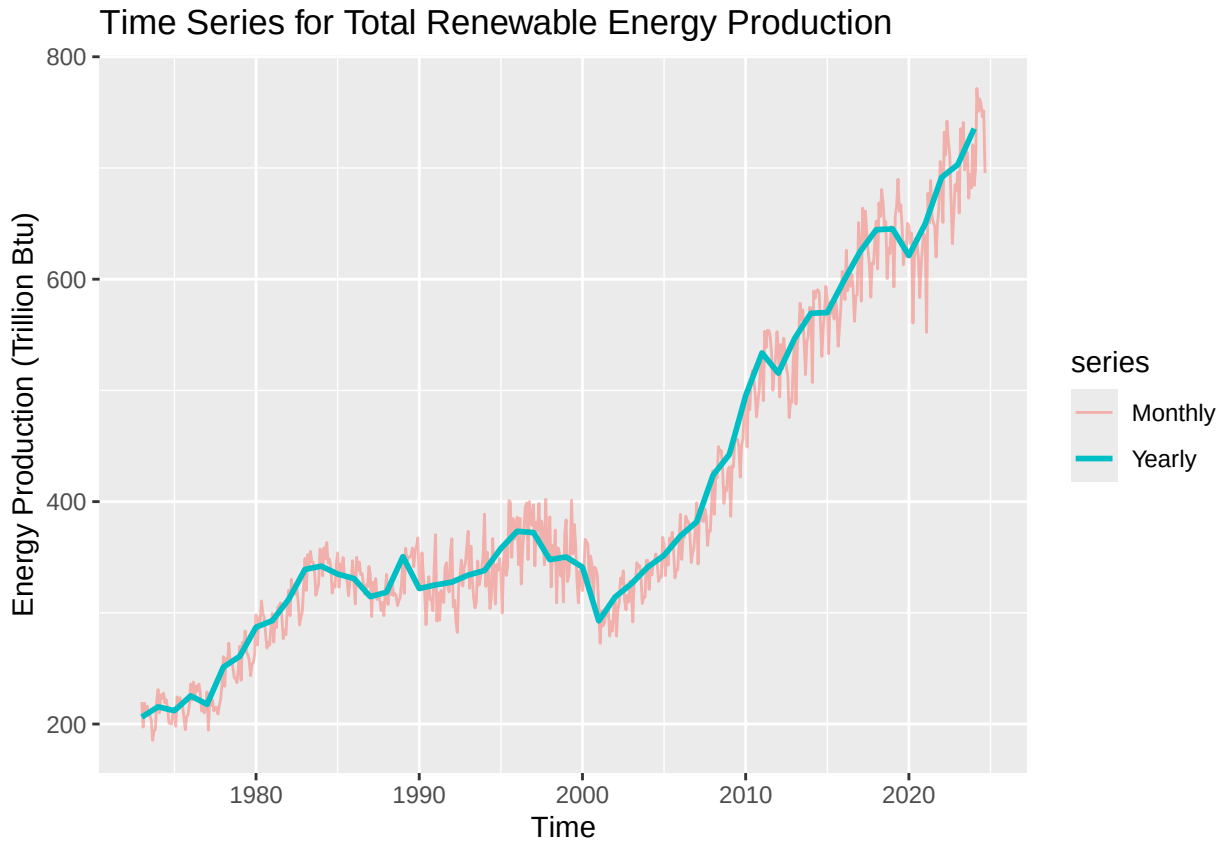
```r
# grouping data in yearly steps instances
power_data_matrix <- matrix(ts_power_data,byrow=FALSE,nrow=12)
```

```
## Warning in matrix(ts_power_data, byrow = FALSE, nrow = 12): data length [621]
## is not a sub-multiple or multiple of the number of rows [12]
```

```r
power_data_matrix[c(10:12), 52] <- NA
power_data_yearly <- colMeans(power_data_matrix, na.rm = TRUE)

ts_power_yearly <- ts(power_data_yearly, start=c(1973,1), frequency=1)

# plotting yearly series
autoplot(ts_power_data, alpha = 0.5, series = "Monthly") +
  autolayer(ts_power_yearly, linewidth = 1, series = "Yearly") +
  ylab("Energy Production (Trillion Btu)") +
  ggtitle("Time Series for Total Renewable Energy Production")
```



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 Q6?

```r
print("Results from Mann-Kendall Test")
```

```
## [1] "Results from Mann-Kendall Test"
```

```r
summary(MannKendall(ts_power_yearly))
```

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

Answer: The p-value is less than 0.05, indicating a significant trend. The Mann-Kendall test score is 1084, a positive number, indicating a strong increasing deterministic trend. They agree with the test results for the monthly series, although the test score value decreased significantly.

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

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

```r
print(cor.test(ts_power_yearly,c(1973:2024),method="spearman"))
```

```
##
##  Spearman's rank correlation rho
##
## data:  ts_power_yearly and c(1973:2024)
## S = 1852, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9209425
```

Answer: The Spearman correlation test showed a significant p-value (less than 0.05) as well, with a rho value of 0.9209, which is close to 1, indicating that the time series for the yearly data shows a positive trend, but not necessarily linear. Although we did not perform Spearman correlation test for the previous monthly data, this result matches with the Mann-Kendall test in terms of suggesting a positive trend for both time series.

```r
print("Results for ADF test")
```

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

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_power_yearly
## Dickey-Fuller = -0.93521, Lag order = 3, p-value = 0.9399
## alternative hypothesis: stationary
```

Answer: The p-value is 0.9399, more than 0.05, meaning that the alternative hypothesis that the time series is stationary (without stochastic trend) is rejected. The test score is -0.93521, with a absolute value close to 1, indicating a unit root and the presence of stochastic trend. This again matches with the monthly data.