ENV 790.30 - Time Series Analysis for Energy Data | Spring 2021

Assignment 4 - Due date 02/25/21

Rajat Khandelwal

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.

Questions

Consider the same data you used for A2 from the spreadsheet "Table_10.1_Renewable_Energy_Production_and_Consumption The data comes from the US Energy Information and Administration and corresponds to the January 2021 Monthly Energy Review.

R packages needed for this assignment:"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.\

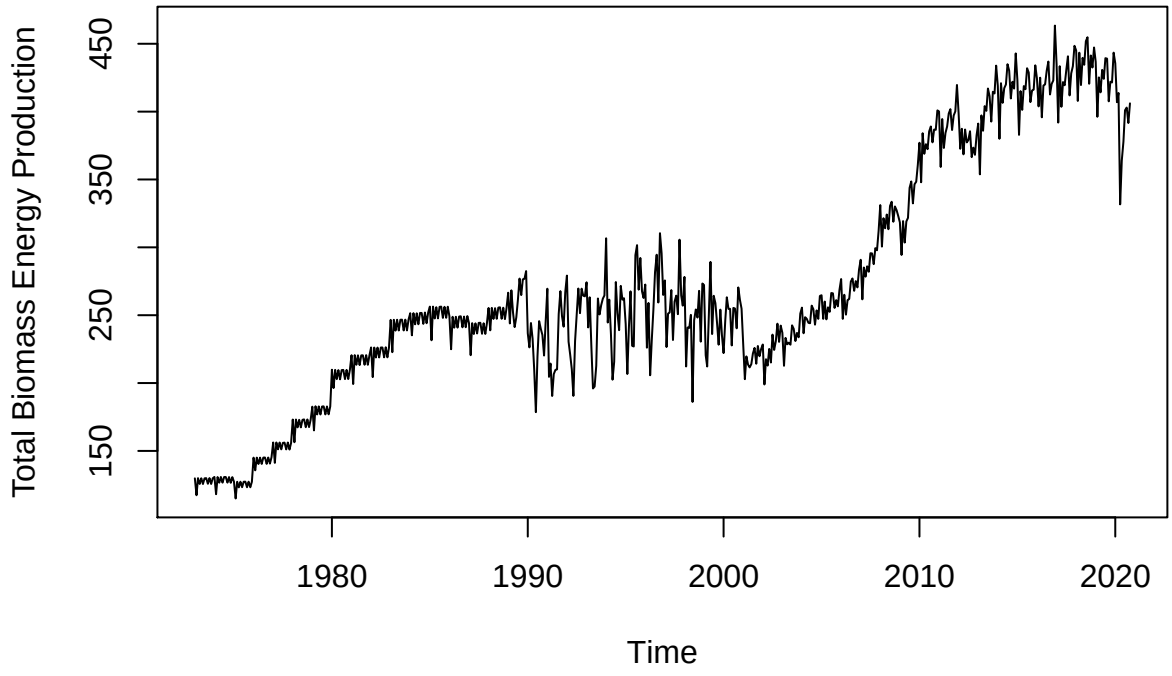
```
#Load/install required package here
library(forecast)
library(tseries)
library(Kendall)
library(readxl)
library(dplyr)
library(lubridate)
library(ggplot2)
library(tidyverse)
```

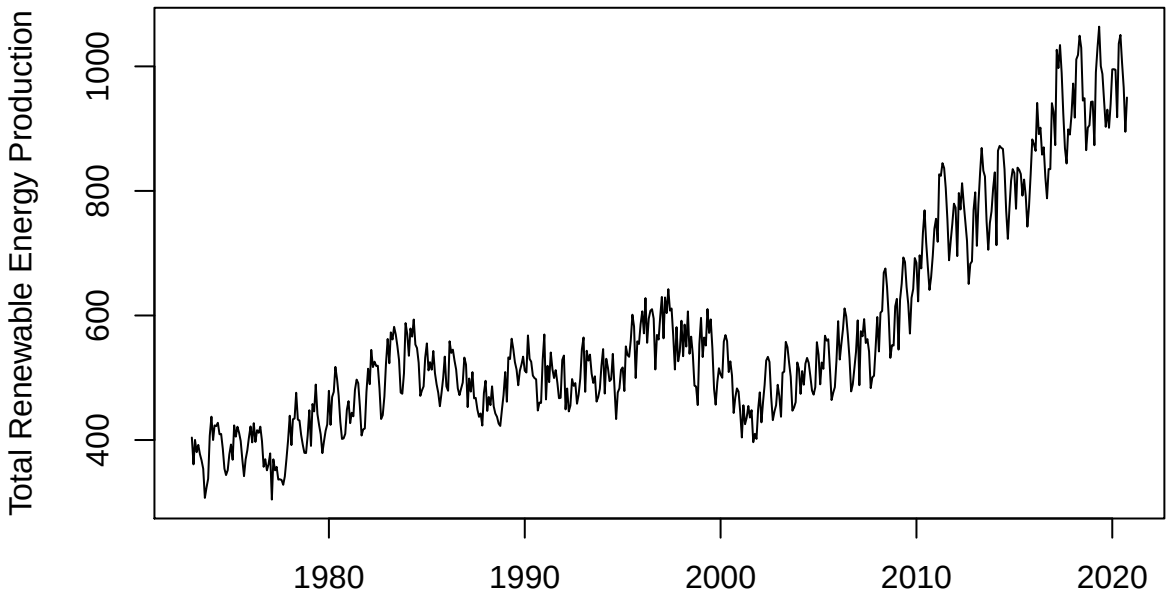
Stochastic Trend and Stationarity Test

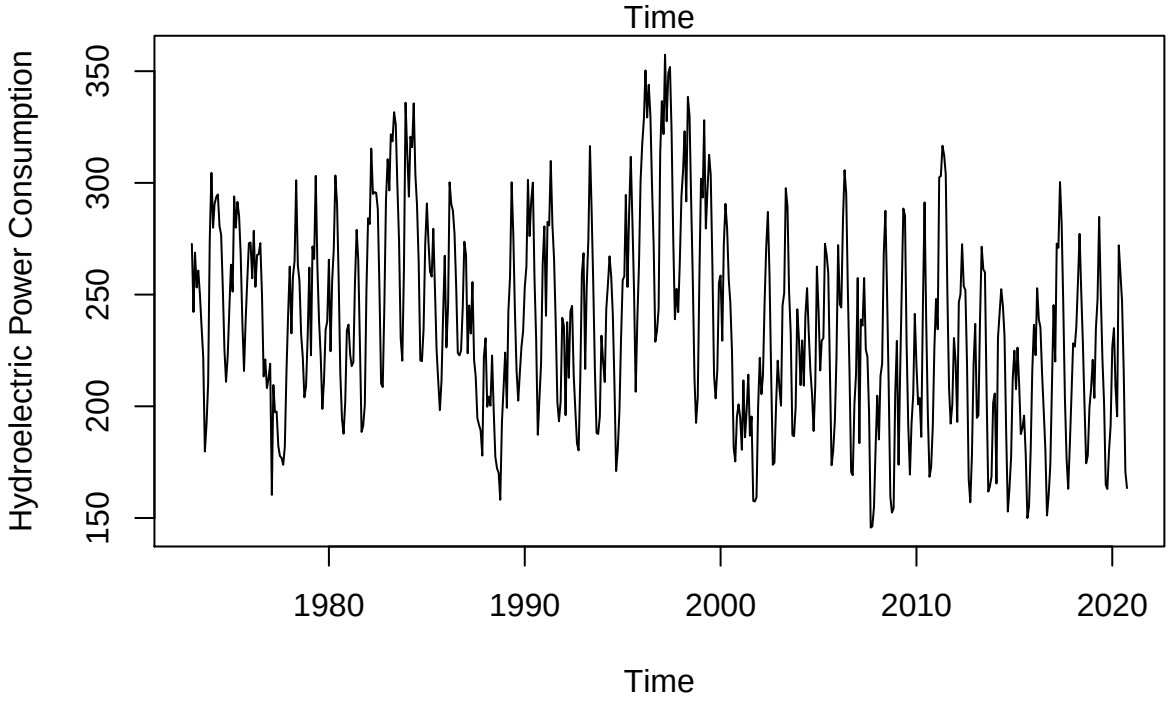
For this part you will once again work only with the following columns: Total Biomass Energy Production, Total Renewable Energy Production, Hydroelectric Power Consumption. Create a data frame structure with these three time series and the Date column. Don't forget to format the date object.

```
#Importing data set
#NOTE: Locally changed name of XLSX file to "REP_Data.xlsx" for easier reference.
data_original <- read_excel("~/Documents/GitHub/ENV790_30_TSA_S2021/Data/REP_Data.xlsx", skip = 10)
data <- data_original %>% select(1,4:6) %>% slice(2:n())
```

```r
#Converting dataframe columns into numeric values
data$`Total Biomass Energy Production` <- as.numeric(data$`Total Biomass Energy Production`)
data$`Total Renewable Energy Production` <- as.numeric(data$`Total Renewable Energy Production`)
data$`Hydroelectric Power Consumption` <- as.numeric(data$`Hydroelectric Power Consumption`)

#Convert time column to date format
data$Month <- as.Date(data$Month , format = "%m/%d/%y")
#data$Month <- month(data$Month)

#Converting dataframe into time-series
data_ts <- ts(data[2:4], start = c(1973,1), end = c(2020,10), frequency = 12)

#Setting parameters for future use
ncols <- ncol(data) - 1
nobs <- nrow(data)

#Plotting the time series
for (i in 1:ncols){
  plot(x = data$Month, y = data_ts[,i], xlab = "Time", ylab =
         colnames(data[i+1]), type = "l")
}
```







Q1

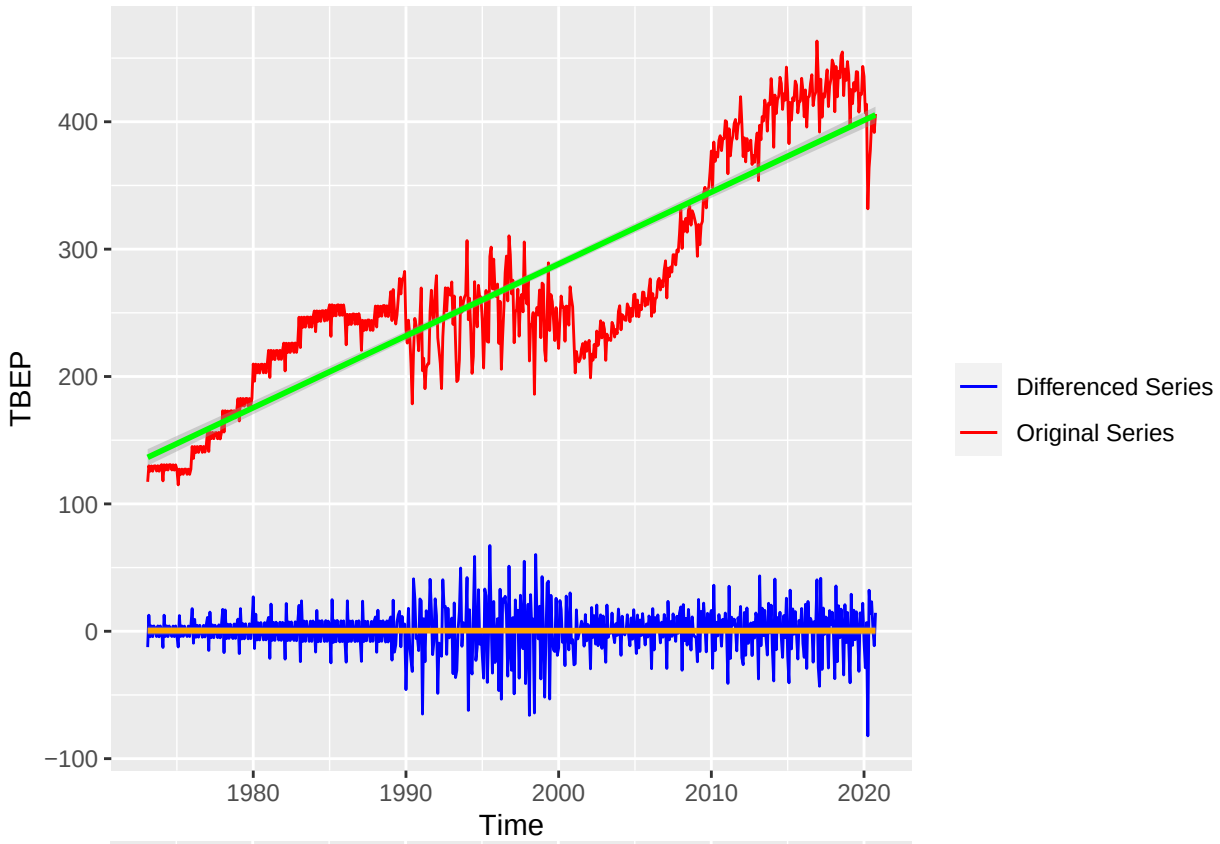
Now let's try to difference these three series using function diff(). Start with the original data from part (b). Try differencing first at lag 1 and plot the remaining series. Did anything change? Do the series still seem to have trend?

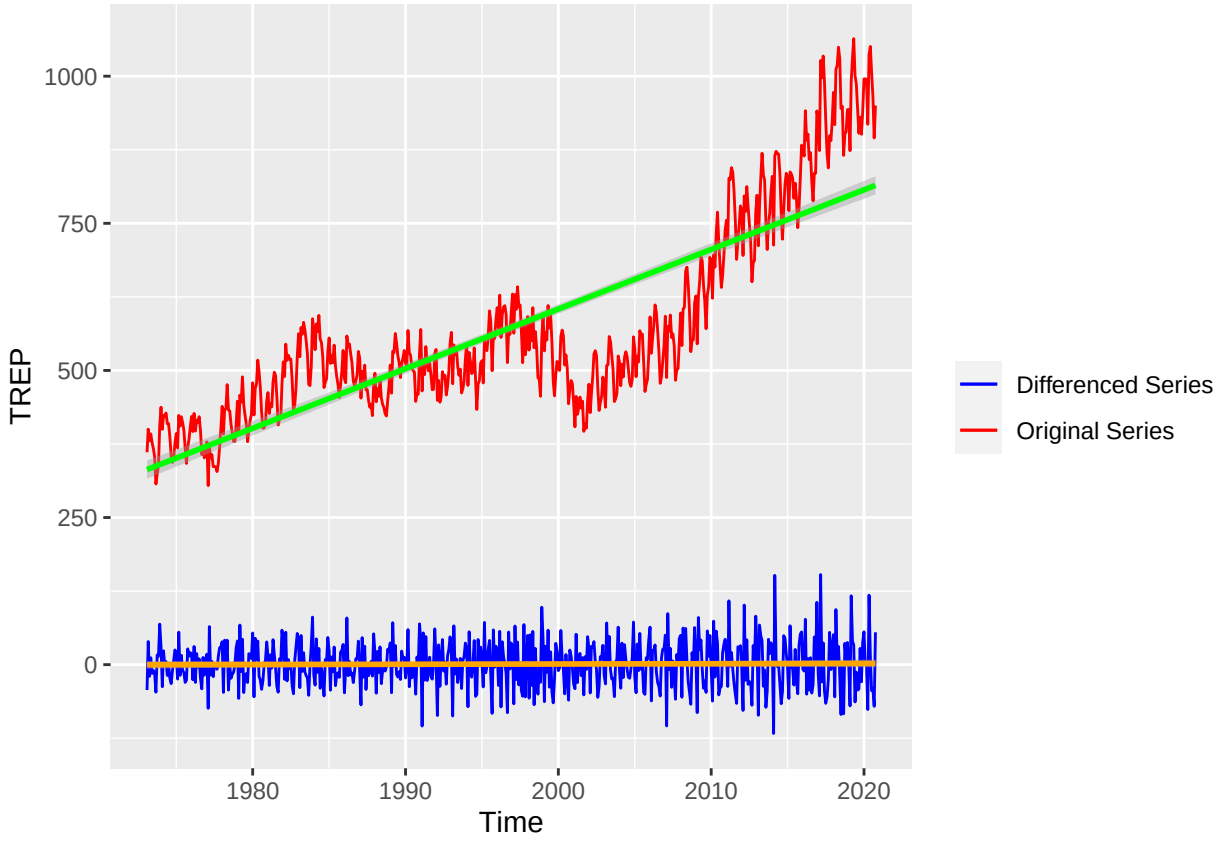
```
#Running ADF test to check for Stochastic Trend
for (i in 1:ncols){
  print(colnames(data[i+1]))
  print(adf.test(data_ts[,i],alternative = "stationary"))
}
```

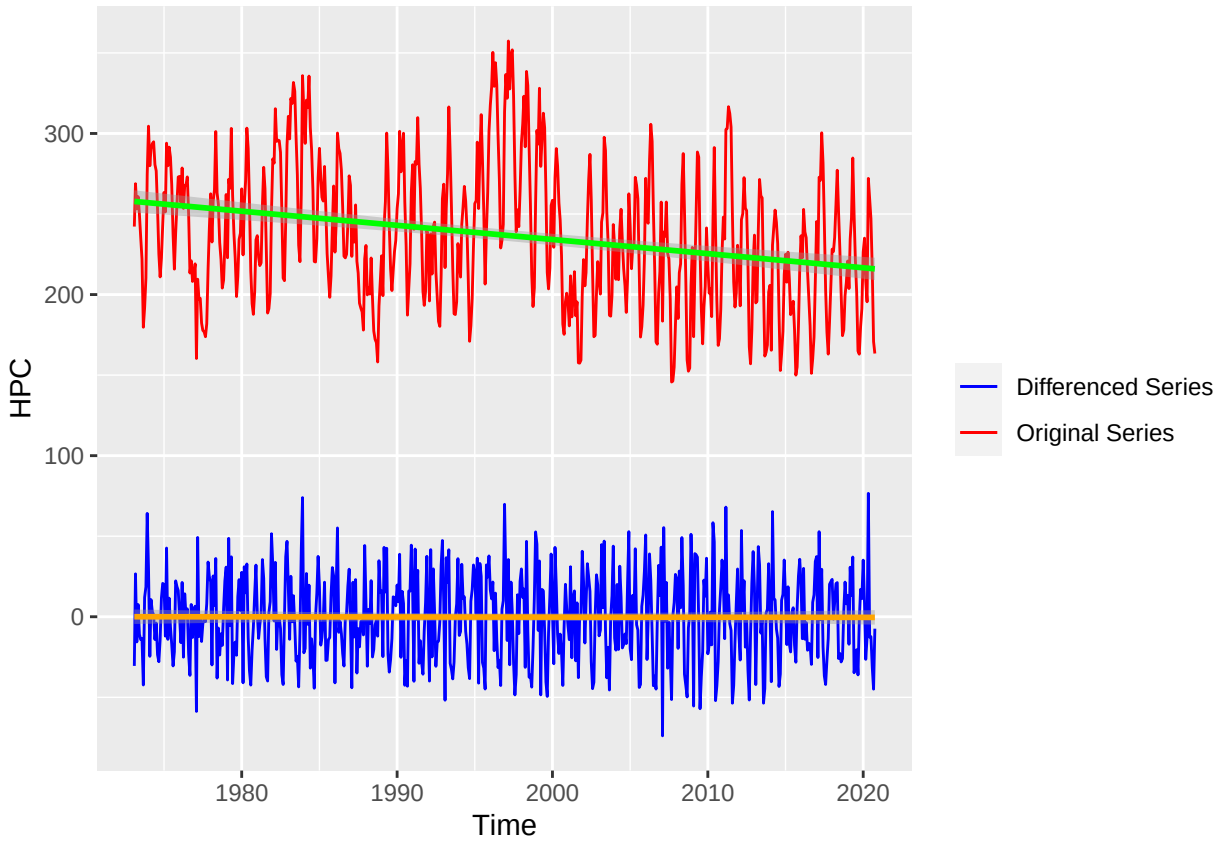
```
## [1] "Total Biomass Energy Production"
##
##  Augmented Dickey-Fuller Test
##
## data:  data_ts[, i]
## Dickey-Fuller = -1.5962, Lag order = 8, p-value = 0.7492
## alternative hypothesis: stationary
##
## [1] "Total Renewable Energy Production"
##
##  Augmented Dickey-Fuller Test
##
## data:  data_ts[, i]
## Dickey-Fuller = -1.5574, Lag order = 8, p-value = 0.7657
## alternative hypothesis: stationary
##
## [1] "Hydroelectric Power Consumption"
##
##  Augmented Dickey-Fuller Test
##
## data:  data_ts[, i]
## Dickey-Fuller = -4.9481, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

As per the ADF test, we **fail to reject the null hypothesis for TBEP and TREP**. We **reject the null hypothesis for HPC**. This means that TBEP and TREP do not have a stochastic trend but they **might** have a deterministic trend. HPC has a stochastc trend. Either way, we will take the difference to remove any sort of trend, if at all it exists.

```
#Creating dataframe with differenced series
data_diff <- data.frame(as.Date(data$Month, format = '%Y-%m-%d'),
          c(NA, diff(data_ts[,1], differences = 1)) , c(NA, diff(data_ts[,2],
          differences = 1)), c(NA, diff(data_ts[,3], differences = 1))) %>%
  na.omit(TBEP)
data_diff <- setNames(data_diff, c("Month", "TBEP","TREP","HPC"))

#Plotting the original and differenced series together
colours <- c("Original Series" = "red", "Differenced Series" = "blue")
for(i in 1:ncols){
print(ggplot(data_diff, aes(x = Month, y = data_diff[,i+1], color =
                            "Differenced Series")) +
        geom_line() +
        xlab("Time") +
        ylab(colnames(data_diff[i+1])) +
        geom_smooth(aes(y=data_diff[,i+1]), method = "lm", color = "orange") +
        geom_line(aes(y = data_ts[2:nobs,i], color = "Original Series"))+
        geom_smooth(aes(y = data_ts[2:nobs,i]), method = "lm", color = "green") +
        scale_colour_manual("", values = c("Original Series"="red",
                                           "Differenced Series"="blue")))
      }
```


TBEP
400
300
200
100
0
−100
1980
1990
2000
2010
2020
Time
Differenced Series
Original Series


TREP
1000
750
500
250
0
1980
1990
2000
2010
2020
Time
Differenced Series
Original Series



After running the diff() function at a lag = 1 to calculate the differenced series, we see that the **trend has been eliminated**. The differenced series (in blue) have a linear model line y = 0, which shows that the differencing the original series removed the trend for all the three series in question.

Q2

Compute Mann-Kendall and Spearman's Correlation Rank Test for each time series. Ask R to print the results. Interpret the results.

```
#Creating list objects to store outputs of decompose and seasadj functions
data_ts_decomp <- vector(mode = "list", length = ncols)
data_ts_decomp_seasadj <- vector(mode = "list", length = ncols)

for (i in 1:ncols){
#Removing seasonality from the data to use Stationarity Tests
  data_ts_decomp[[i]] <- decompose(data_ts[,i], "additive")
  data_ts_decomp_seasadj[[i]] <- seasadj(data_ts_decomp[[i]])

#Stationarity Tests
  print(colnames(data[i+1]))
  print("Spearman's Rank Correlation Test")
  print(cor.test(data_ts_decomp_seasadj[[i]], array(1:574, dim = c(574,1)),
                 method = "spearman"))
  print("Mann-Kendall's Test")
  print(MannKendall(data_ts_decomp_seasadj[[i]]))
}
```

```
## [1] "Total Biomass Energy Production"
```

```
## [1] "Spearman's Rank Correlation Test"
##
##  Spearman's rank correlation rho
##
## data:  data_ts_decomp_seasadj[[i]] and array(1:574, dim = c(574, 1))
## S = 4098238, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.8699788
##
## [1] "Mann-Kendall's Test"
## tau = 0.718, 2-sided pvalue =< 2.22e-16
## [1] "Total Renewable Energy Production"
## [1] "Spearman's Rank Correlation Test"
##
##  Spearman's rank correlation rho
##
## data:  data_ts_decomp_seasadj[[i]] and array(1:574, dim = c(574, 1))
## S = 4829142, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.8467901
##
## [1] "Mann-Kendall's Test"
## tau = 0.689, 2-sided pvalue =< 2.22e-16
## [1] "Hydroelectric Power Consumption"
## [1] "Spearman's Rank Correlation Test"
##
##  Spearman's rank correlation rho
##
## data:  data_ts_decomp_seasadj[[i]] and array(1:574, dim = c(574, 1))
## S = 44725066, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.4189526
##
## [1] "Mann-Kendall's Test"
## tau = -0.273, 2-sided pvalue =< 2.22e-16
```

The test coefficients (rho, tau) for HPC are negative, indicating a **negative coorelation (decreasing trend over time**) between this series and time. These coefficients are positive for TBEP and TREP, showing a **positive correlation (increasing trend over time)** between these two series and time.

To **test for the significance** of these correlations, we refer to the **hypothesis tests**. For all the three series, we **reject the null hypothesis that the series carry a deterministic trend** due to the extremely small p-values observed in the Spearman's Rank Correlation Test and Mann-Kendall test. Therefore, we can conclude that **all three series show a trend**.

Decomposing the series

For this part you will work only with the following columns: Solar Energy Consumption and Wind Energy Consumption.

Q3

Create a data frame structure with these two time series only and the Date column. Drop the rows with *Not Available* and convert the columns to numeric. You can use filtering to eliminate the initial rows or conver to numeric and then use the drop_na() function. If you are familiar with pipes for data wrangling, try using it!

```
data_renewable <- data_original %>% select(1,8,9) %>% slice(2:n())
#Converting to numeric data type to drop NA data later
data_renewable$`Solar Energy Consumption` <- as.numeric(data_renewable$`Solar Energy Consumption`)
data_renewable$`Wind Energy Consumption` <- as.numeric(data_renewable$`Wind Energy Consumption`)
#Converting Month column to type 'date'
data_renewable$Month <- as.Date(data_renewable$Month, format = "%m/%d/%y")
#Dropping NA data
data_renewable <- drop_na(data_renewable)
head(data_renewable)
```

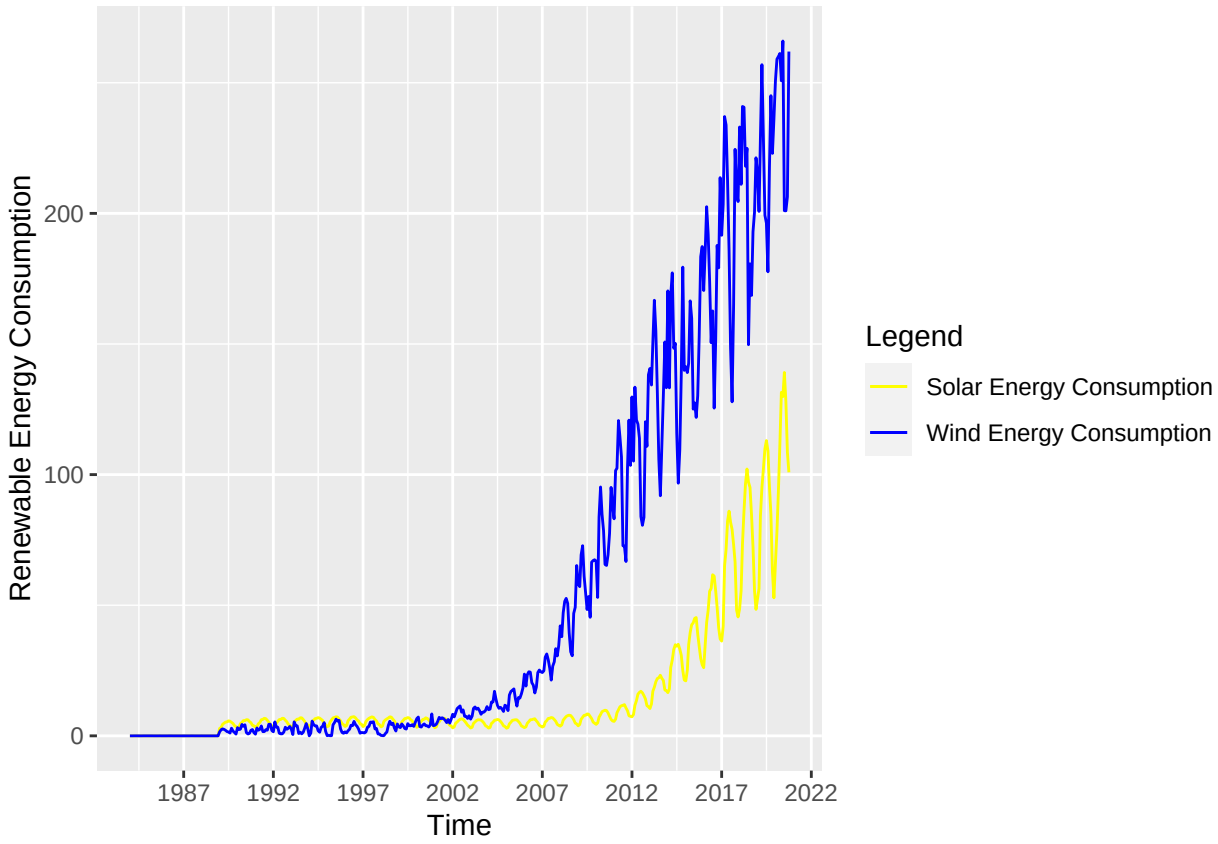
```
## # A tibble: 6 x 3
##   Month      `Solar Energy Consumption` `Wind Energy Consumption`
##   <date>                          <dbl>                     <dbl>
## 1 1984-01-01                     -0.001                     0
## 2 1984-02-01                      0.001                     0.002
## 3 1984-03-01                      0.002                     0.002
## 4 1984-04-01                      0.003                     0.006
## 5 1984-05-01                      0.007                     0.008
## 6 1984-06-01                      0.01                      0.006
```

Q4

Plot the Solar and Wind energy consumption over time using ggplot. Explore the function scale_x_date() on ggplot and see if you can change the x axis to improve your plot. Hint: use *scale_x_date(date_breaks = "5 years", date_labels = "%Y")")*

Try changing the color of the wind series to blue. Hint: use *color = "blue"*

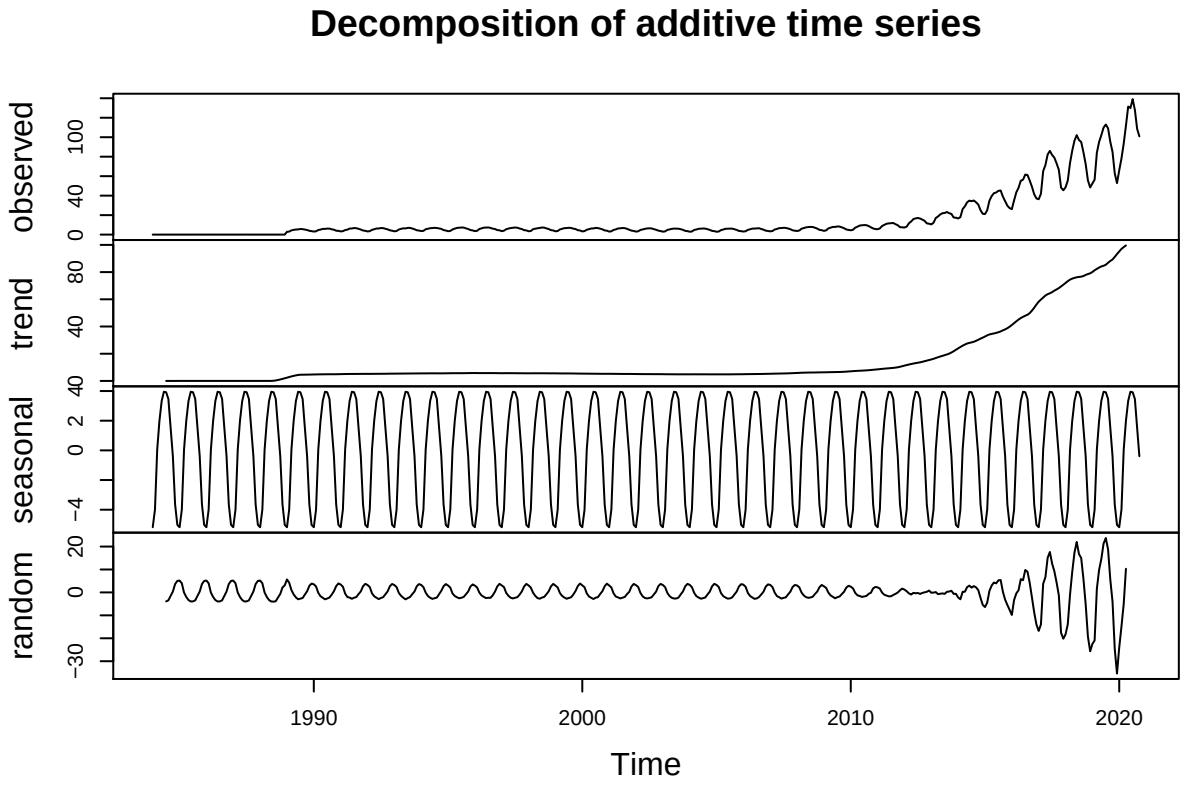
```
ggplot(data_renewable, aes(x = Month)) +
  geom_line(aes(y = `Solar Energy Consumption`, color = "Solar Energy Consumption")) +
  geom_line(aes(y = `Wind Energy Consumption`, color = "Wind Energy Consumption")) +
  scale_colour_manual("Legend",breaks = c("Solar Energy Consumption",
                                          "Wind Energy Consumption"),
                      values = c("yellow", "blue")) +
  xlab("Time") +
  ylab("Renewable Energy Consumption") +
  scale_x_date(date_breaks ="5 years", date_labels = "%Y",
               limits = as.Date(c("1984-01-01","2020-10-01")))
```



Q5

Transform wind and solar series into a time series object and apply the decompose function on them using the additive option. What can you say about the trend component? What about the random component? Does the random component look random? Or does it appear to still have some seasonality on it?

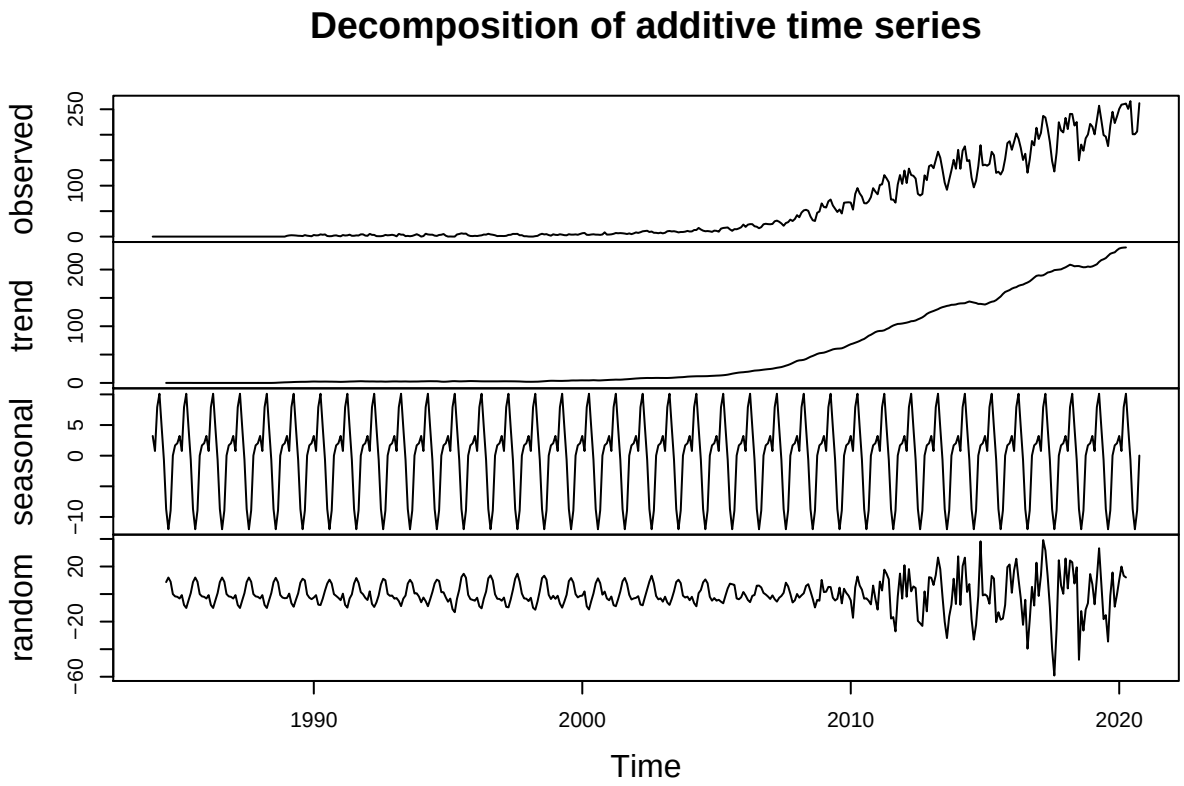
```
data_renewable_ts <- ts(data_renewable[,2:3], start = c(1984,01), frequency = 12)
solar_decompose_add <- decompose(data_renewable_ts[,1], "additive")
wind_decompose_add <- decompose(data_renewable_ts[,2], "additive")
plot(solar_decompose_add)
```



Solar Energy Consumption

Trend: We observe an increasing trend towards the end of the series.
Seasonality: We observe strong seasonality as is expected of Solar Energy Consumption data. We expect higher production (and hence consumption) in the summer months, lower in the rainy/winter seasons.
Random: The random component does not look random, rather it looks to have some sort of seasonality which increases in amplitude towards the end of the series.

```
plot(wind_decompose_add)
```



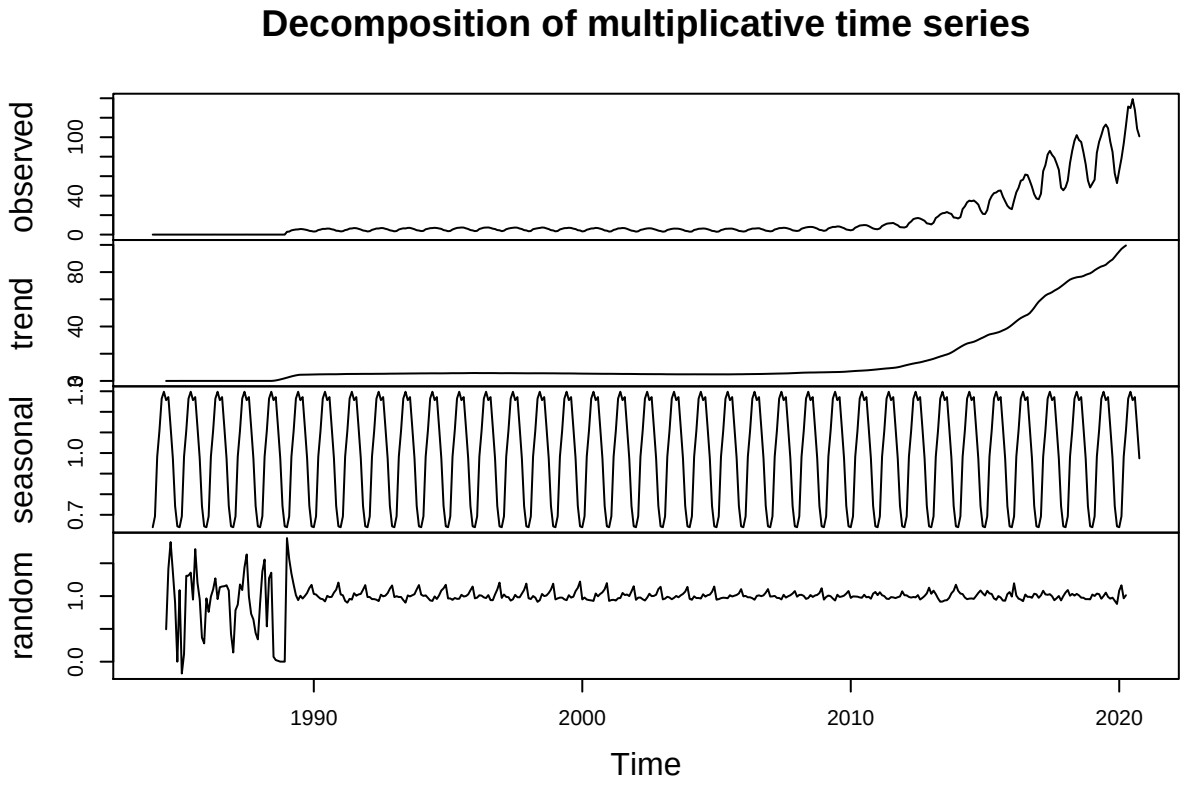
Wind Energy Consumption

Trend: Shows an increasing trend towards the end of ther series. However, this starts much earlier than the trend line for the Solar Energy Consumption.
Seasonality: Strong seasonality is observed.
Random: The random component shows seasonality.

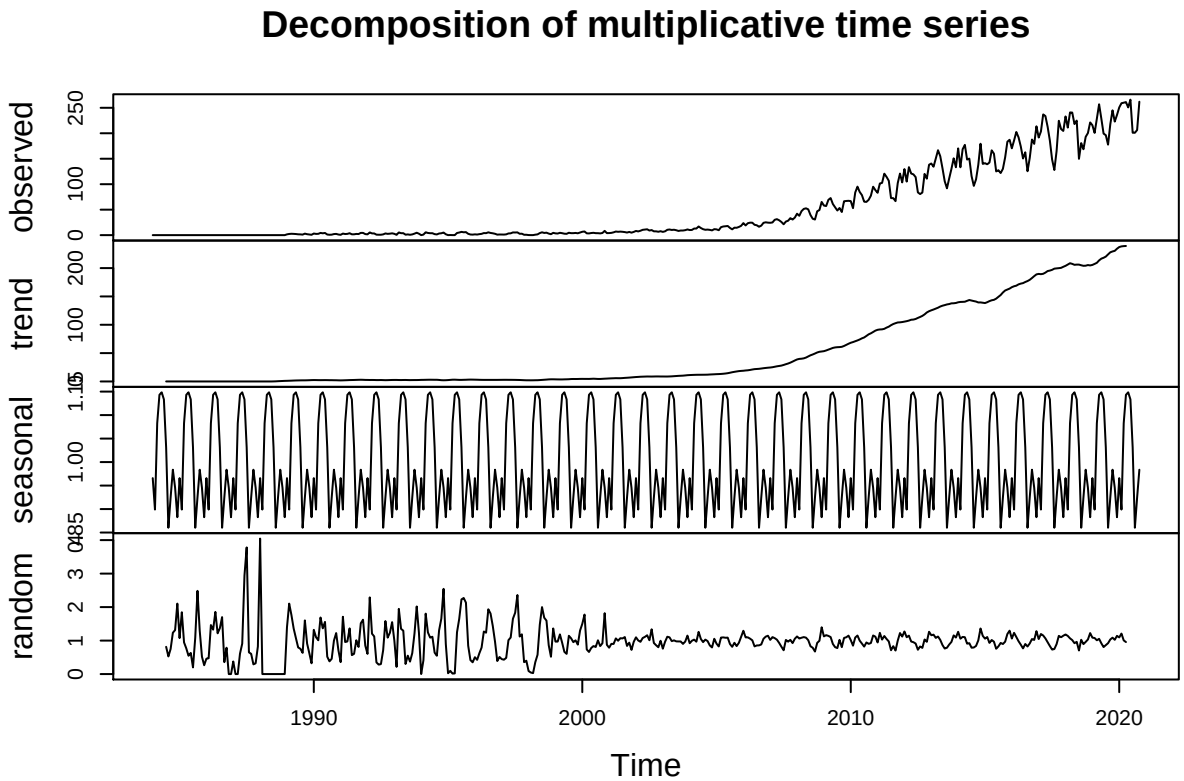
Q6

Use the decompose function again but now change the type of the seasonal component from additive to multiplicative. What happened to the random component this time?

```
solar_decompose_mul <- decompose(data_renewable_ts[,1], "multiplicative")
wind_decompose_mul <- decompose(data_renewable_ts[,2], "multiplicative")
plot(solar_decompose_mul)
```




```
plot(wind_decompose_mul)
```



The random component for both the series, Solar and Wind Energy Consumption **does not show seasonality** anymore and appears to be truly random.

Q7

When fitting a model to this data, do you think you need all the historical data? Think about the date from 90s and early 20s. Are there any information from those year we might need to forecast the next six months of Solar and/or Wind consumption. Explain your response.

A7: No, I believe that taking all the historical data would be misrepresentative while forecasting for the future. Current trends in the data would get diminished due to the influence of data from the late 80s and early 90s, when **renewable energy consumption** was quite low and showed big variations on a monthly basis. Further, 2020 was an outlier year, as due to **COVID-19** safety measures, a large section of the population stayed at home, driving up electricity consumption. This is reflected in the data as we can observe a sudden peak for renewable energy consumption in 2020 as compared to previous years.

Therefore, while forecasting data for the next six months, it would be **not advisable** to use very old (90s) and extremely recent (2020) data as these would lead to inaccurate forecasts.