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

Assignment 4 - Due date 02/25/21

Xueying Feng

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_Consumpt
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
#install.packages("readxl")
library("readxl")
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
library(ggplot2)
#install.packages("forecast")
library(forecast)
```

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

```r
#install.packages("tseries")
library(tseries)
library(Kendall)

#install.packages("outliers")
library(outliers)
#install.packages("tidyverse")
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------------------- t

## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## v purrr   0.3.4

## -- Conflicts ------------------------------------------------------------------------------ tidyvers
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

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.

```r
#Importing data set

MonthlyData <- read_excel("../Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xls
                          sheet = 1, skip = 9)

# number of obs
nobsv <- nrow(MonthlyData)


# Select columns: Total Biomass Energy Production, Total Renewable Energy Production, Hydroelectric Pow
MonthlyData_subset<- MonthlyData[2:nobsv, c(1, 4, 5, 6)]

# Checking data
head(MonthlyData_subset)
```

```
## # A tibble: 6 x 4
##   Month               `Total Biomass Ene~ `Total Renewable E~ `Hydroelectric Po~
##   <dttm>              <chr>               <chr>               <chr>
## 1 1973-01-01 00:00:00 129.787             403.981             272.703
## 2 1973-02-01 00:00:00 117.338             360.9               242.199
```

```
## 3 1973-03-01 00:00:00 129.938             400.161             268.81
## 4 1973-04-01 00:00:00 125.636             380.47              253.185
## 5 1973-05-01 00:00:00 129.834             392.141             260.77
## 6 1973-06-01 00:00:00 125.611             377.232             249.859
```

```
str(MonthlyData_subset)
```

```
## tibble [574 x 4] (S3: tbl_df/tbl/data.frame)
##  $ Month                            : POSIXct[1:574], format: "1973-01-01" "1973-02-01" ...
##  $ Total Biomass Energy Production  : chr [1:574] "129.787" "117.338" "129.938" "125.636" ...
##  $ Total Renewable Energy Production: chr [1:574] "403.981" "360.9" "400.161" "380.47" ...
##  $ Hydroelectric Power Consumption  : chr [1:574] "272.703" "242.199" "268.81" "253.185" ...
```

```
# Change tbl_df to data frame
MonthlyData_subset = as.data.frame(MonthlyData_subset)
str(MonthlyData_subset)
```

```
## 'data.frame':    574 obs. of  4 variables:
##  $ Month                            : POSIXct, format: "1973-01-01" "1973-02-01" ...
##  $ Total Biomass Energy Production  : chr  "129.787" "117.338" "129.938" "125.636" ...
##  $ Total Renewable Energy Production: chr  "403.981" "360.9" "400.161" "380.47" ...
##  $ Hydroelectric Power Consumption  : chr  "272.703" "242.199" "268.81" "253.185" ...
```

```
# number of col
ncoln<- ncol(MonthlyData_subset)-1

# Change column names
#colnames(MonthlyData_subset)[1] <- "Date"
colnames(MonthlyData_subset)=c("Date","Biomass","Renewable","Hydroelectric")
str(MonthlyData_subset)
```

```
## 'data.frame':    574 obs. of  4 variables:
##  $ Date         : POSIXct, format: "1973-01-01" "1973-02-01" ...
##  $ Biomass      : chr  "129.787" "117.338" "129.938" "125.636" ...
##  $ Renewable    : chr  "403.981" "360.9" "400.161" "380.47" ...
##  $ Hydroelectric: chr  "272.703" "242.199" "268.81" "253.185" ...
```

```
# change character format to numeric format
MonthlyData_subset[,2:4] <- sapply(MonthlyData_subset[,2:4],as.numeric)
str(MonthlyData_subset)
```

```
## 'data.frame':    574 obs. of  4 variables:
##  $ Date         : POSIXct, format: "1973-01-01" "1973-02-01" ...
##  $ Biomass      : num  130 117 130 126 130 ...
##  $ Renewable    : num  404 361 400 380 392 ...
##  $ Hydroelectric: num  273 242 269 253 261 ...
```

```
# Create a data frame structure with these three time series
# From Jan 1973 to Oct 2020 as a time series object
MonthlyData_subset_ts <- ts(MonthlyData_subset[,2:4], frequency = 12, start = c(1973, 1, 1), end = c(20
str(MonthlyData_subset_ts)
```

```
##  Time-Series [1:574, 1:3] from 1973 to 2021: 130 117 130 126 130 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:3] "Biomass" "Renewable" "Hydroelectric"
```

```r
MonthlyData_subset_ts <- as.ts(MonthlyData_subset_ts[1:574,])
MyDate <- as.Date(MonthlyData_subset$Date)

#create new df
MonthlyData_new <- cbind.data.frame(MyDate, MonthlyData_subset_ts)
str(MonthlyData_new)
```

```
## 'data.frame':    574 obs. of  4 variables:
##  $ MyDate       : Date, format: "1973-01-01" "1973-02-01" ...
##  $ Biomass      : num  130 117 130 126 130 ...
##  $ Renewable    : num  404 361 400 380 392 ...
##  $ Hydroelectric: num  273 242 269 253 261 ...
```

```r
class(MonthlyData_new)
```

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

```r
head(MonthlyData_new)
```

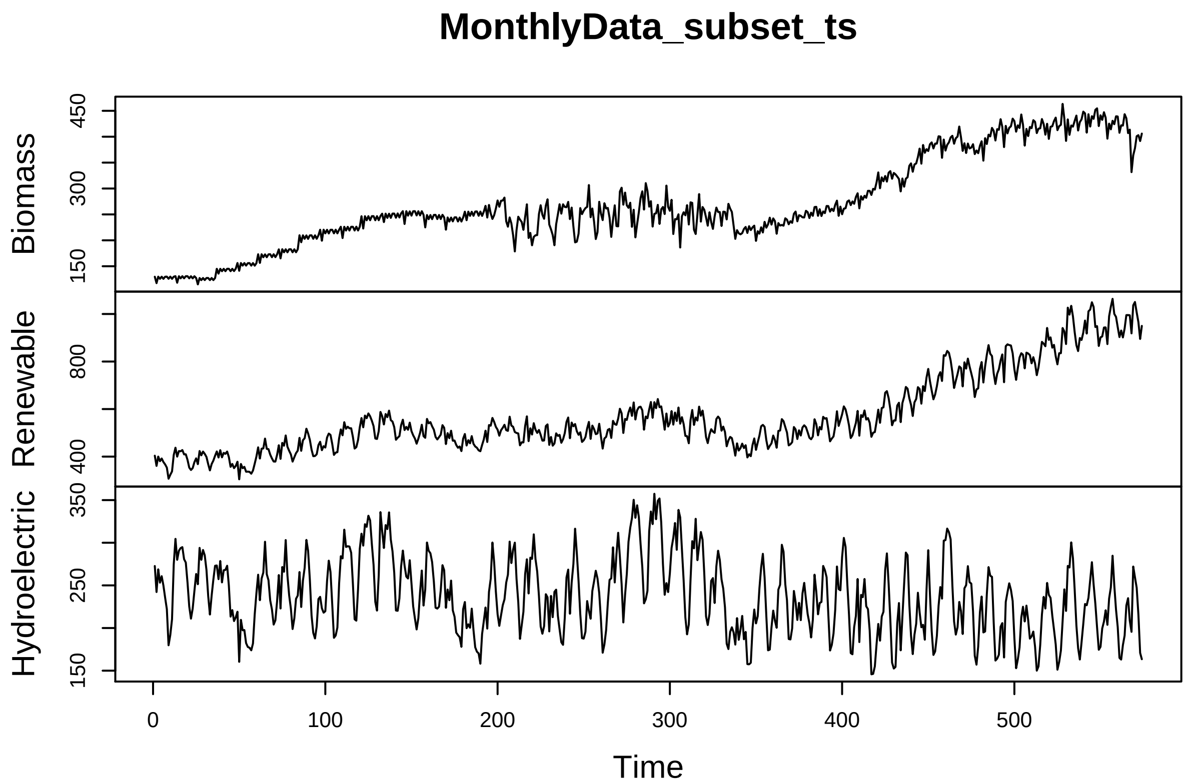
```
##       MyDate Biomass Renewable Hydroelectric
## 1 1973-01-01 129.787   403.981       272.703
## 2 1973-02-01 117.338   360.900       242.199
## 3 1973-03-01 129.938   400.161       268.810
## 4 1973-04-01 125.636   380.470       253.185
## 5 1973-05-01 129.834   392.141       260.770
## 6 1973-06-01 125.611   377.232       249.859
```

Q1

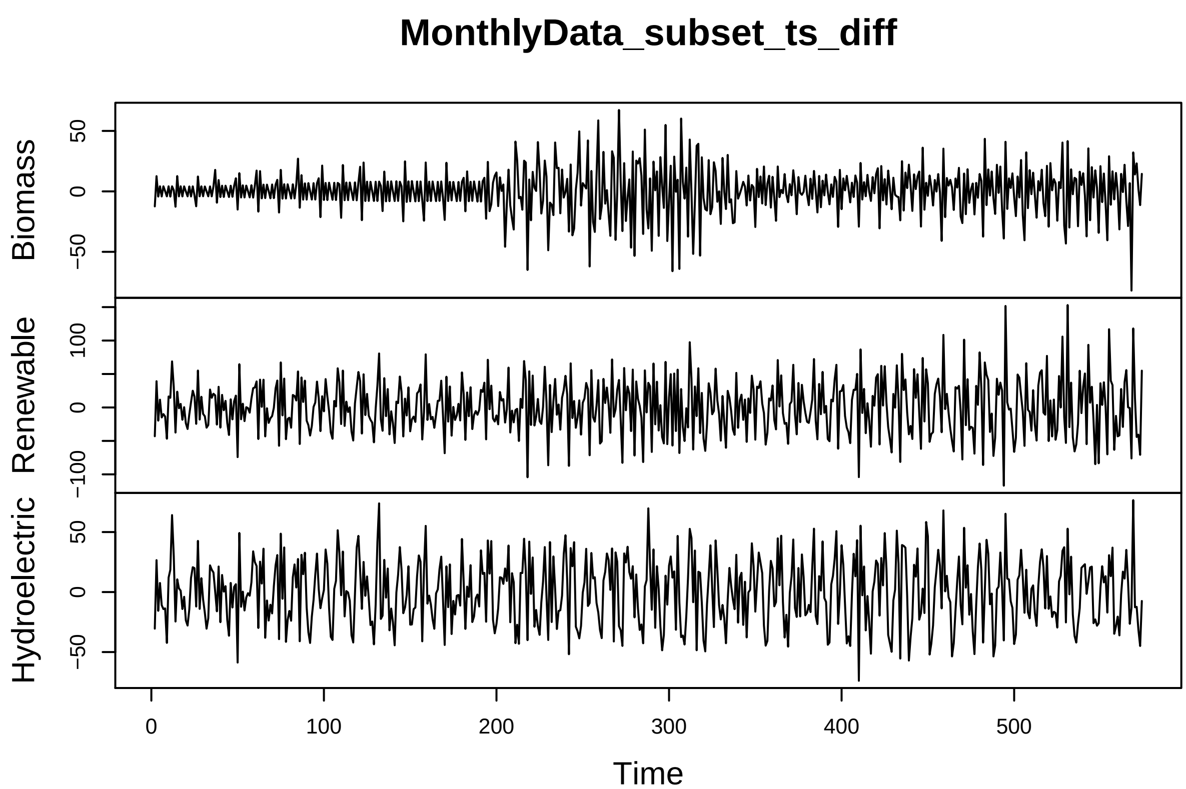
Now let's try to difference these three series using function diff(). Start with the original data from part (b). Try differencing first at lag 1 and plot the remaining series. Did anything change? Do the series still seem to have trend?

```r
MonthlyData_subset_ts_diff <- diff(MonthlyData_subset_ts, lag = 1, differences = 1)

#plot
plot(MonthlyData_subset_ts)
```




```
plot(MonthlyData_subset_ts_diff)
```



Q2

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

```
# Change MonthlyData_subset_ts_diff to data frame
df_diff<- as.data.frame(MonthlyData_subset_ts_diff)


#Add the new series(frame?) to our data frame
df_MonthlyData_full <-
  MonthlyData_subset %>%
  cbind(BioDiff = c(NA,as.numeric(df_diff$Biomass))) %>%
  cbind(RenewDiff = c(NA,as.numeric(df_diff$Renewable))) %>%
  cbind(HydroDiff = c(NA,as.numeric(df_diff$Hydroelectric))) %>%
  na.omit(df_MonthlyData_full)

head(df_MonthlyData_full)
```

```
##         Date Biomass Renewable Hydroelectric BioDiff RenewDiff HydroDiff
## 2 1973-02-01 117.338   360.900       242.199 -12.449   -43.081   -30.504
## 3 1973-03-01 129.938   400.161       268.810  12.600    39.261    26.611
## 4 1973-04-01 125.636   380.470       253.185  -4.302   -19.691   -15.625
## 5 1973-05-01 129.834   392.141       260.770   4.198    11.671     7.585
## 6 1973-06-01 125.611   377.232       249.859  -4.223   -14.909   -10.911
## 7 1973-07-01 129.787   367.325       235.670   4.176    -9.907   -14.189
```

```
#Since I have seasonal data I cannot use the simple MannKendall()
#another example of functions that need a ts object
ts_MonthlyData_full <- ts(df_MonthlyData_full[,2:7],
                          frequency = 12,
                          start = c(year(df_MonthlyData_full$Date[1]),
                                    month(df_MonthlyData_full$Date[1])))
str(ts_MonthlyData_full)
```

```
##  Time-Series [1:573, 1:6] from 1973 to 2021: 117 130 126 130 126 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:6] "Biomass" "Renewable" "Hydroelectric" "BioDiff" ...
```

```
head(ts_MonthlyData_full)
```

```
##          Biomass Renewable Hydroelectric BioDiff RenewDiff HydroDiff
## Feb 1973 117.338   360.900       242.199 -12.449   -43.081   -30.504
## Mar 1973 129.938   400.161       268.810  12.600    39.261    26.611
## Apr 1973 125.636   380.470       253.185  -4.302   -19.691   -15.625
## May 1973 129.834   392.141       260.770   4.198    11.671     7.585
## Jun 1973 125.611   377.232       249.859  -4.223   -14.909   -10.911
## Jul 1973 129.787   367.325       235.670   4.176    -9.907   -14.189
```

```r
#Mann-Kendall Correlation
for(ColNum in 1:6){
  SMKtest_MonthlyData <- SeasonalMannKendall(ts_MonthlyData_full[,ColNum])
  print("Results for Seasonal Mann Kendall")
  print(summary(SMKtest_MonthlyData))
}
```

```
## [1] "Results for Seasonal Mann Kendall"
## Score =  9829 , Var(Score) = 149601
## denominator =  13395
## tau = 0.734, 2-sided pvalue =< 2.22e-16
## NULL
## [1] "Results for Seasonal Mann Kendall"
## Score =  9433 , Var(Score) = 149601
## denominator =  13395
## tau = 0.704, 2-sided pvalue =< 2.22e-16
## NULL
## [1] "Results for Seasonal Mann Kendall"
## Score =  -3855 , Var(Score) = 149601
## denominator =  13395
## tau = -0.288, 2-sided pvalue =< 2.22e-16
## NULL
## [1] "Results for Seasonal Mann Kendall"
## Score =  568 , Var(Score) = 149600
## denominator =  13394.5
## tau = 0.0424, 2-sided pvalue =0.14196
## NULL
## [1] "Results for Seasonal Mann Kendall"
## Score =  111 , Var(Score) = 149601
## denominator =  13395
## tau = 0.00829, 2-sided pvalue =0.77413
## NULL
## [1] "Results for Seasonal Mann Kendall"
## Score =  -257 , Var(Score) = 149601
## denominator =  13395
## tau = -0.0192, 2-sided pvalue =0.5064
## NULL
```

```r
for(ColNum in 1:6){
  #Group data in yearly steps instances
  ts_MonthlyData_full_matrix <- matrix(ts_MonthlyData_full[,ColNum],byrow=FALSE,nrow=12)
  MonthlyData_full_yearly <- colMeans(ts_MonthlyData_full_matrix)

  #library(dplyr)  #move this to package chunk later
  Year <- c(year(first(df_MonthlyData_full$Date)):year(last(df_MonthlyData_full$Date)))

  MonthlyData_full_new_yearly <- data.frame(Year, MonthlyData_full_yearly)
  head(MonthlyData_full_new_yearly)
  tail(MonthlyData_full_new_yearly)
  str(MonthlyData_full_new_yearly)

  print("Results from Spearman Correlation")
  SpCor_MonthlyData=cor.test(MonthlyData_full_yearly,Year,method="spearman")
```

```
  print(SpCor_MonthlyData)
}
```

```
## Warning in matrix(ts_MonthlyData_full[, ColNum], byrow = FALSE, nrow = 12): data
## length [573] is not a sub-multiple or multiple of the number of rows [12]

## 'data.frame':    48 obs. of  2 variables:
##  $ Year                   : int  1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 ...
##  $ MonthlyData_full_yearly: num  128 128 126 144 155 ...
## [1] "Results from Spearman Correlation"
##
##  Spearman's rank correlation rho
##
## data:  MonthlyData_full_yearly and Year
## S = 2264, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.8771168

## Warning in matrix(ts_MonthlyData_full[, ColNum], byrow = FALSE, nrow = 12): data
## length [573] is not a sub-multiple or multiple of the number of rows [12]

## 'data.frame':    48 obs. of  2 variables:
##  $ Year                   : int  1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 ...
##  $ MonthlyData_full_yearly: num  370 391 393 390 356 ...
## [1] "Results from Spearman Correlation"
##
##  Spearman's rank correlation rho
##
## data:  MonthlyData_full_yearly and Year
## S = 2524, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.8630048

## Warning in matrix(ts_MonthlyData_full[, ColNum], byrow = FALSE, nrow = 12): data
## length [573] is not a sub-multiple or multiple of the number of rows [12]

## 'data.frame':    48 obs. of  2 variables:
##  $ Year                   : int  1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 ...
##  $ MonthlyData_full_yearly: num  241 261 264 244 198 ...
## [1] "Results from Spearman Correlation"
##
##  Spearman's rank correlation rho
##
## data:  MonthlyData_full_yearly and Year
## S = 27594, p-value = 0.0003795
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.4977204
```

```
## Warning in matrix(ts_MonthlyData_full[, ColNum], byrow = FALSE, nrow = 12): data
## length [573] is not a sub-multiple or multiple of the number of rows [12]
```

```
## 'data.frame':    48 obs. of  2 variables:
##  $ Year                  : int  1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 ...
##  $ MonthlyData_full_yearly: num  0.085 -0.295 1.482 0.931 1.409 ...
## [1] "Results from Spearman Correlation"
##
##  Spearman's rank correlation rho
##
## data:  MonthlyData_full_yearly and Year
## S = 20776, p-value = 0.386
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.1276596
```

```
## Warning in matrix(ts_MonthlyData_full[, ColNum], byrow = FALSE, nrow = 12): data
## length [573] is not a sub-multiple or multiple of the number of rows [12]
```

```
## 'data.frame':    48 obs. of  2 variables:
##  $ Year                  : int  1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 ...
##  $ MonthlyData_full_yearly: num  2.79 -3.73 2.42 -3.6 5.04 ...
## [1] "Results from Spearman Correlation"
##
##  Spearman's rank correlation rho
##
## data:  MonthlyData_full_yearly and Year
## S = 16046, p-value = 0.3808
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.1290708
```

```
## Warning in matrix(ts_MonthlyData_full[, ColNum], byrow = FALSE, nrow = 12): data
## length [573] is not a sub-multiple or multiple of the number of rows [12]
```

```
## 'data.frame':    48 obs. of  2 variables:
##  $ Year                  : int  1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 ...
##  $ MonthlyData_full_yearly: num  2.65 -3.428 0.829 -4.521 3.622 ...
## [1] "Results from Spearman Correlation"
##
##  Spearman's rank correlation rho
##
## data:  MonthlyData_full_yearly and Year
## S = 20330, p-value = 0.4829
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## -0.103452
```

```r
#cor(MonthlyData_full_yearly,Year,method="spearman")
```

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!

```r
#Importing data set
MonthlyData2 <- read_excel("../Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xls
                           sheet = 1, skip = 9)
# number of obs
nobsv <- nrow(MonthlyData2)

# Select columns: Solar Energy Consumption and Wind Energy Consumption
MonthlyData_subset2<- MonthlyData2[2:nobsv, c(1, 8, 9)]

# change'Not Available' to "NA"
MonthlyData_subset2[] <- lapply(MonthlyData_subset2, gsub, pattern='Not Available', replacement='NA')

# Checking data
head(MonthlyData_subset2)
```

```
## # A tibble: 6 x 3
##   Month      `Solar Energy Consumption` `Wind Energy Consumption`
##   <chr>      <chr>                      <chr>
## 1 1973-01-01 NA                         NA
## 2 1973-02-01 NA                         NA
## 3 1973-03-01 NA                         NA
## 4 1973-04-01 NA                         NA
## 5 1973-05-01 NA                         NA
## 6 1973-06-01 NA                         NA
```

```r
str(MonthlyData_subset2)
```

```
## tibble [574 x 3] (S3: tbl_df/tbl/data.frame)
##  $ Month                   : chr [1:574] "1973-01-01" "1973-02-01" "1973-03-01" "1973-04-01" ...
##  $ Solar Energy Consumption: chr [1:574] "NA" "NA" "NA" "NA" ...
##  $ Wind Energy Consumption : chr [1:574] "NA" "NA" "NA" "NA" ...
```

```r
# Change column names
#colnames(MonthlyData_subset2)[1] <- "Date"
colnames(MonthlyData_subset2) <- c("Date","Solar","Wind")
MonthlyData_subset2$Date <- as.Date(MonthlyData_subset2$Date)
str(MonthlyData_subset2)
```

```
## tibble [574 x 3] (S3: tbl_df/tbl/data.frame)
##  $ Date : Date[1:574], format: "1973-01-01" "1973-02-01" ...
##  $ Solar: chr [1:574] "NA" "NA" "NA" "NA" ...
##  $ Wind : chr [1:574] "NA" "NA" "NA" "NA" ...
```

```
# change character format to numeric format
MonthlyData_subset2[,2:3] <- sapply(MonthlyData_subset2[,2:3],as.numeric)
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
# Drop NA
#library(dplyr)
MonthlyData_subset_new <- na.omit(MonthlyData_subset2)
str(MonthlyData_subset_new)
```

```
## tibble [442 x 3] (S3: tbl_df/tbl/data.frame)
##  $ Date : Date[1:442], format: "1984-01-01" "1984-02-01" ...
##  $ Solar: num [1:442] -0.001 0.001 0.002 0.003 0.007 0.01 0.003 0.009 0.01 0.007 ...
##  $ Wind : num [1:442] 0 0.002 0.002 0.006 0.008 0.006 0.005 0.003 0.005 0.009 ...
##  - attr(*, "na.action")= 'omit' Named int [1:132] 1 2 3 4 5 6 7 8 9 10 ...
##   ..- attr(*, "names")= chr [1:132] "1" "2" "3" "4" ...
```

```
head(MonthlyData_subset_new)
```

```
## # A tibble: 6 x 3
##   Date        Solar  Wind
##   <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
```

```
tail(MonthlyData_subset_new)
```

```
## # A tibble: 6 x 3
##   Date       Solar  Wind
##   <date>     <dbl> <dbl>
## 1 2020-05-01  131.  251.
## 2 2020-06-01  130.  266.
## 3 2020-07-01  139.  201.
## 4 2020-08-01  128.  201.
## 5 2020-09-01  109.  206.
## 6 2020-10-01  101.  262.
```

```
# Create a data frame structure with these three time series
# From Jan 1984 as a time series object
MonthlyData_subset_ts2 <- ts(MonthlyData_subset_new[,2:3], frequency = 12,
                             start = c(1984,1,1),end = c(2020,10,1))
str(MonthlyData_subset_ts2)
```

```
## Time-Series [1:442, 1:2] from 1984 to 2021: -0.001 0.001 0.002 0.003 0.007 0.01 0.003 0.009 0.01 0.0
## - attr(*, "dimnames")=List of 2
##  ..$ : NULL
##  ..$ : chr [1:2] "Solar" "Wind"
```

```
MyDate2 <- as.Date(MonthlyData_subset_new$Date)
head(MyDate2)
```

```
## [1] "1984-01-01" "1984-02-01" "1984-03-01" "1984-04-01" "1984-05-01"
## [6] "1984-06-01"
```

```
#create new df
MonthlyData_new2 <- cbind.data.frame(MyDate2, MonthlyData_subset_ts2)
str(MonthlyData_new2)
```

```
## 'data.frame':    442 obs. of  3 variables:
##  $ MyDate2: Date, format: "1984-01-01" "1984-02-01" ...
##  $ Solar  : num  -0.001 0.001 0.002 0.003 0.007 0.01 0.003 0.009 0.01 0.007 ...
##  $ Wind   : num  0 0.002 0.002 0.006 0.008 0.006 0.005 0.003 0.005 0.009 ...
```

```
class(MonthlyData_new2)
```

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

```
head(MonthlyData_new2)
```

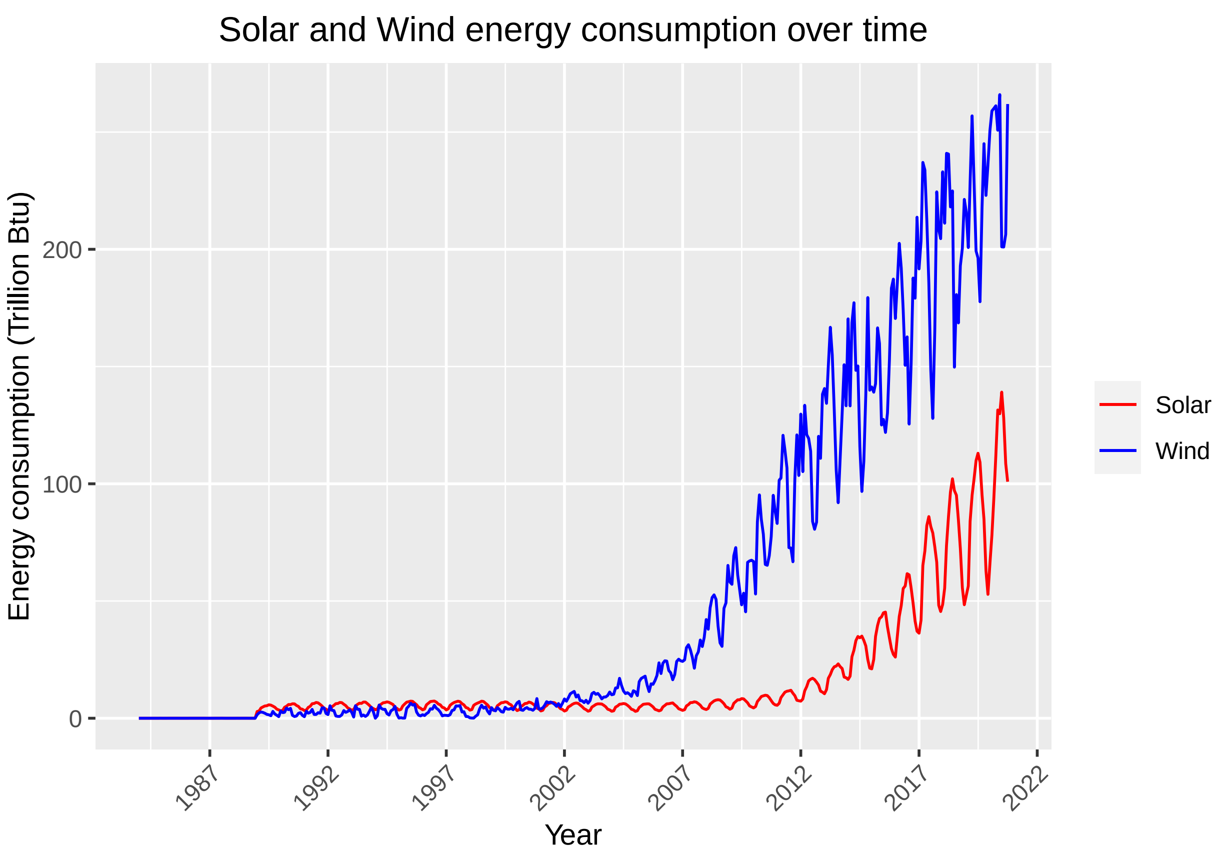
```
##      MyDate2  Solar  Wind
## 1 1984-01-01 -0.001 0.000
## 2 1984-02-01  0.001 0.002
## 3 1984-03-01  0.002 0.002
## 4 1984-04-01  0.003 0.006
## 5 1984-05-01  0.007 0.008
## 6 1984-06-01  0.010 0.006
```

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(MonthlyData_new2) +
  geom_line(aes(MyDate2, y=Solar, colour = "Solar")) +
  geom_line(aes(MyDate2, y = Wind, colour = "Wind")) +
  ggtitle("Solar and Wind energy consumption over time") +
  xlab("Year") +
  ylab("Energy consumption (Trillion Btu)") +
  scale_x_date(date_breaks = "5 years", date_labels = "%Y") +
  scale_colour_manual("", values = c("Solar" = "red", "Wind"= "blue")) +
  theme(axis.text.x = element_text(angle = 45,  hjust = 1),
        plot.title = element_text(hjust=0.5))
```



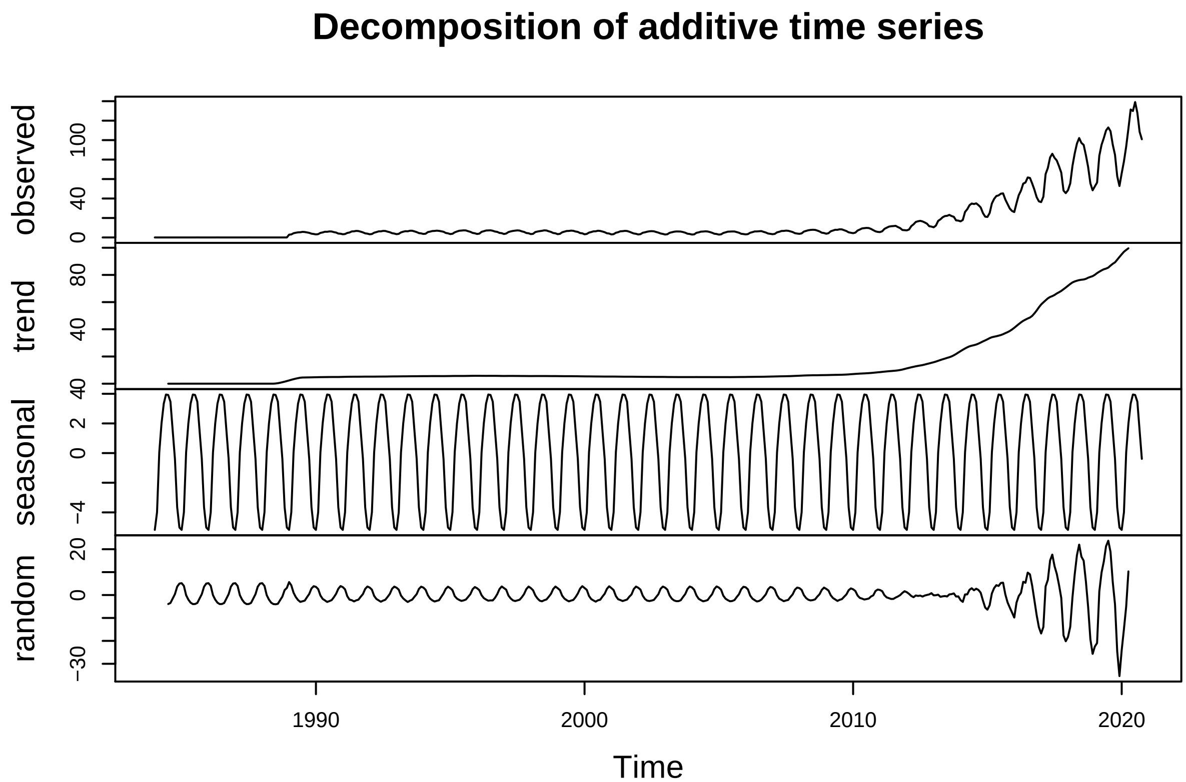
Q5

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?

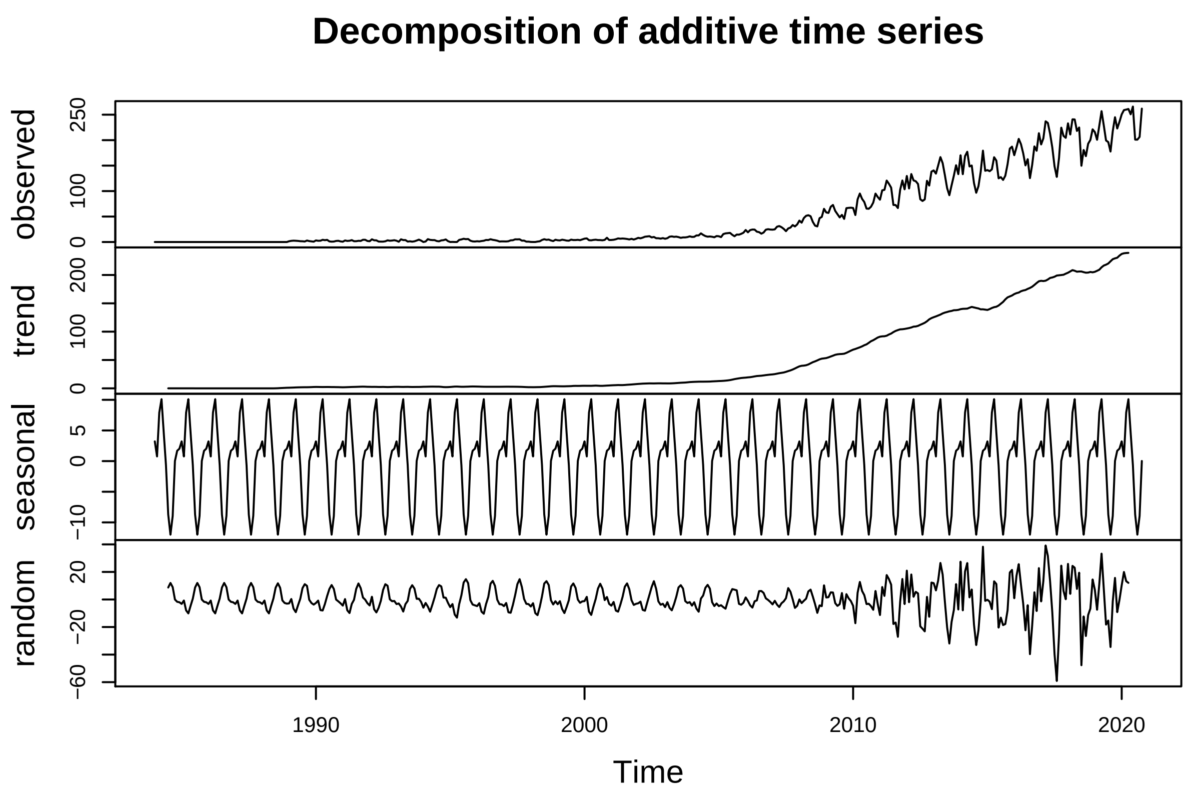
```
str(MonthlyData_subset_ts2)
```

```
##  Time-Series [1:442, 1:2] from 1984 to 2021: -0.001 0.001 0.002 0.003 0.007 0.01 0.003 0.009 0.01 0.0
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:2] "Solar" "Wind"
```

```
#Using R decompose function
decompose_Solar_ts2 <- decompose(MonthlyData_subset_ts2[,"Solar"],"additive")
plot(decompose_Solar_ts2)
```




```
decompose_Wind_ts2 <- decompose(MonthlyData_subset_ts2[,"Wind"],"additive")
plot(decompose_Wind_ts2)
```



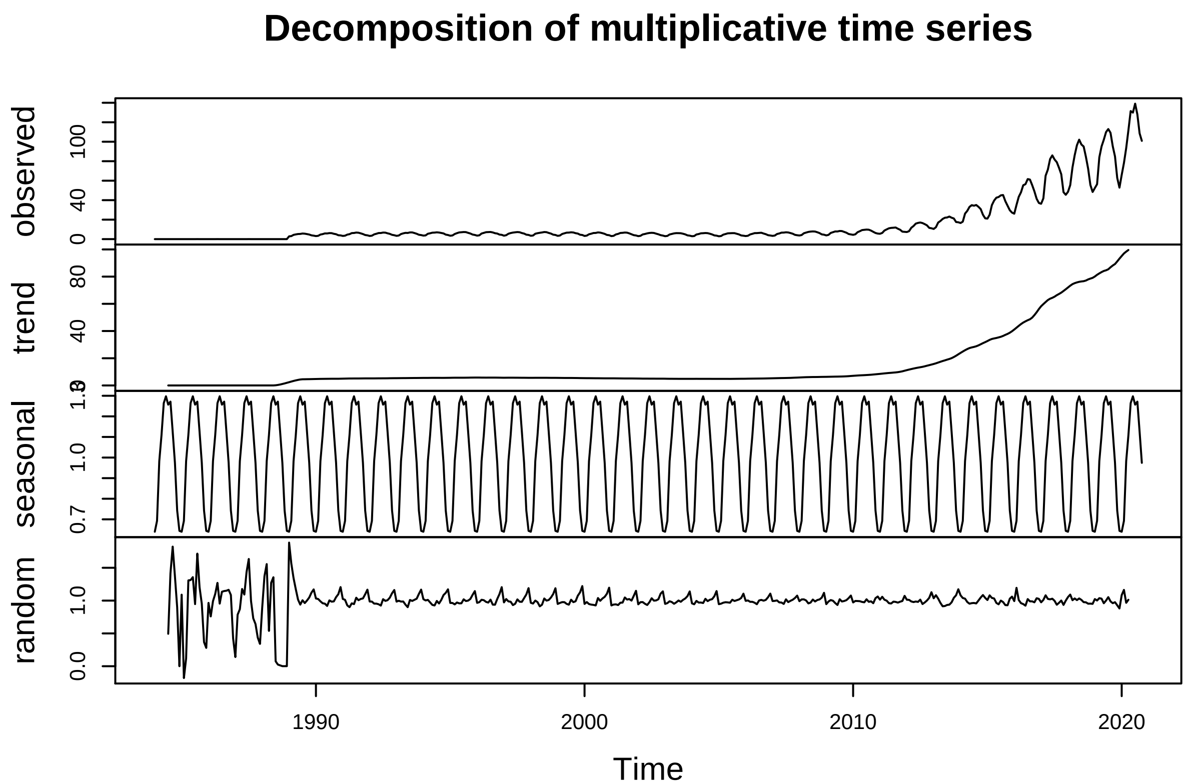
>The trend component of both of them is an increasing pattern. The random component is not that kind of

random and it is a regularly repeating pattern before 2015 in Solar dataset and before 2010 in Wind dataset. So there still are some seasonality on that.

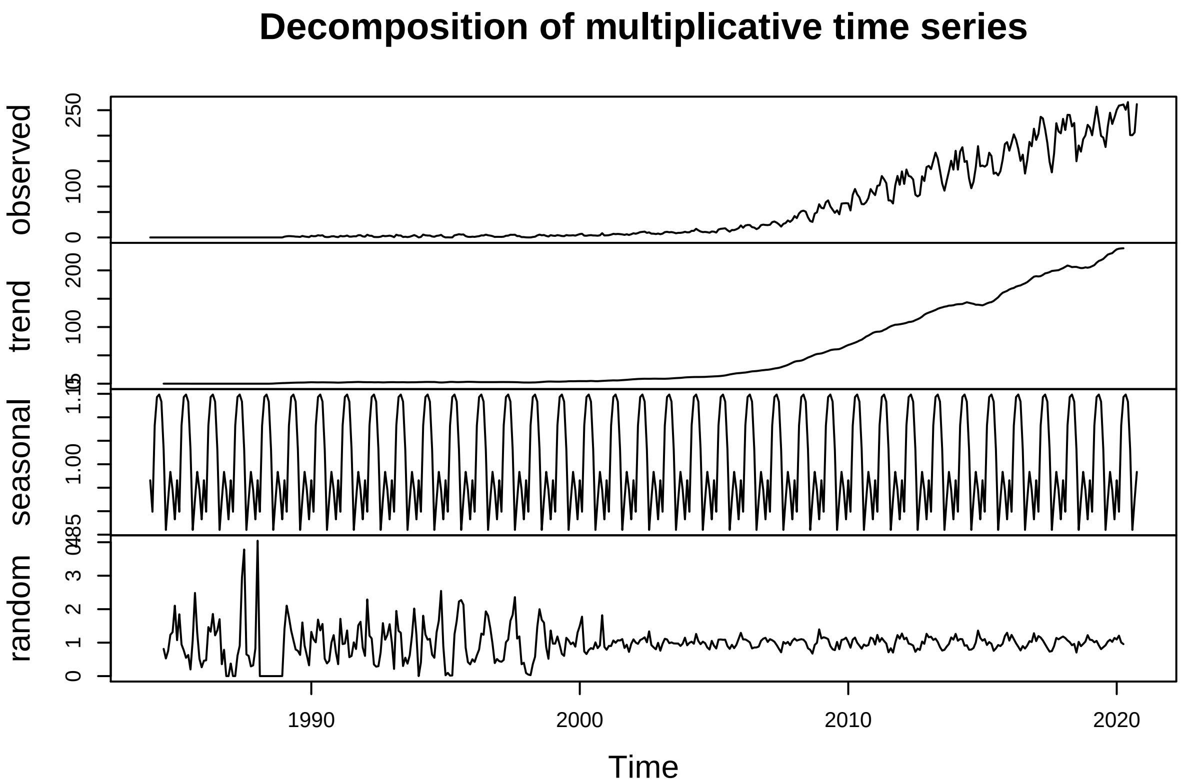
Q6

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?

```
decompose_Solar_ts2 <- decompose(MonthlyData_subset_ts2[,"Solar"],"multiplicative")
plot(decompose_Solar_ts2)
```




```
#par(cex.lab=1.2)
decompose_Wind_ts2 <- decompose(MonthlyData_subset_ts2[,"Wind"],"multiplicative")
plot(decompose_Wind_ts2)
```



>The random outputs are different from “additive” ones. The random pattern occur bedore 1990 for Solar, and occur before 2000 in Wind.

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. >If there is a consistent pattern for all historical data, then it is useful to fit the model and predict the future trend. >For Solar and/or Wind consumption dataset, I think the data after 1990 in Solar and after 2000 in Wind should be use to forecast the next six month comsumption.