ENV 790.30 - Time Series Analysis for Energy Data | Spring 2022

Assignment 5 - Due date 02/28/22

Abbhijith Hari Gopal

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_A05_Sp22.Rmd"). Submit this pdf using Sakai.

R packages needed for this assignment are listed below. 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(xlsx)
```

```
## Warning: package 'xlsx' was built under R version 4.0.5
```

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.0.5
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.5
```

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

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.0.5
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```r
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.0.5
```

```r
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.5
```

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

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

```r
library(tidyverse)  #load this package so yon clean the data frame using pipes
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
## v purrr   0.3.4
```

```
## Warning: package 'tibble' was built under R version 4.0.5
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
## Warning: package 'purrr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
## Warning: package 'stringr' was built under R version 4.0.5
```

```
## Warning: package 'forcats' was built under R version 4.0.5
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## 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()
```

Decomposing Time Series

Consider the same data you used for A04 from the spreadsheet "Table_10.1_Renewable_Energy_Production_and_Consump
The data comes from the US Energy Information and Administration and corresponds to the January 2021
Monthly Energy Review.

```
#Importing data set - using xlsx package
energy_data <- read.xlsx(file="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.

#Now let's extract the column names from row 11 only
read_col_names <- read.xlsx(file="./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Sourc

colnames(energy_data) <- read_col_names
head(energy_data)
```

```
##        Month Wood Energy Production Biofuels Production
## 1 1973-01-01                129.630       Not Available
## 2 1973-02-01                117.194       Not Available
## 3 1973-03-01                129.763       Not Available
## 4 1973-04-01                125.462       Not Available
## 5 1973-05-01                129.624       Not Available
## 6 1973-06-01                125.435       Not Available
##   Total Biomass Energy Production Total Renewable Energy Production
## 1                         129.787                           403.981
## 2                         117.338                           360.900
## 3                         129.938                           400.161
## 4                         125.636                           380.470
## 5                         129.834                           392.141
## 6                         125.611                           377.232
##   Hydroelectric Power Consumption Geothermal Energy Consumption
## 1                         272.703                         1.491
## 2                         242.199                         1.363
## 3                         268.810                         1.412
## 4                         253.185                         1.649
## 5                         260.770                         1.537
## 6                         249.859                         1.763
##   Solar Energy Consumption Wind Energy Consumption Wood Energy Consumption
## 1            Not Available           Not Available                 129.630
## 2            Not Available           Not Available                 117.194
## 3            Not Available           Not Available                 129.763
## 4            Not Available           Not Available                 125.462
## 5            Not Available           Not Available                 129.624
## 6            Not Available           Not Available                 125.435
##   Waste Energy Consumption Biofuels Consumption
## 1                    0.157        Not Available
## 2                    0.144        Not Available
## 3                    0.176        Not Available
## 4                    0.174        Not Available
## 5                    0.210        Not Available
## 6                    0.176        Not Available
##   Total Biomass Energy Consumption Total Renewable Energy Consumption
## 1                          129.787                            403.981
## 2                          117.338                            360.900
## 3                          129.938                            400.161
```

```
## 4                           125.636                            380.470
## 5                           129.834                            392.141
## 6                           125.611                            377.232
```

```
nobs=nrow(energy_data)
nvar=ncol(energy_data)
```

Q1

For this assignment you will work only with the following columns: Solar Energy Consumption and Wind Energy Consumption. 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 convert to numeric and then use the drop_na() function. If you are familiar with pipes for data wrangling, try using it!

```
date_of_interest = energy_data[, 1]
Solar_Wind = energy_data[, 8:9]
head(date_of_interest, 5) #As a check
```

```
## [1] "1973-01-01" "1973-02-01" "1973-03-01" "1973-04-01" "1973-05-01"
```

```
head(Solar_Wind, 5) #As a check
```

```
##   Solar Energy Consumption Wind Energy Consumption
## 1            Not Available           Not Available
## 2            Not Available           Not Available
## 3            Not Available           Not Available
## 4            Not Available           Not Available
## 5            Not Available           Not Available
```

```
#converting to numeric
Solar_Wind$`Solar Energy Consumption` = as.numeric(Solar_Wind$`Solar Energy Consumption`)
```

```
## Warning: NAs introduced by coercion
```

```
Solar_Wind$`Wind Energy Consumption` = as.numeric(Solar_Wind$`Wind Energy Consumption`)
```

```
## Warning: NAs introduced by coercion
```

```
class(Solar_Wind$`Solar Energy Consumption`) #As a check
```

```
## [1] "numeric"
```

```
class(Solar_Wind$`Wind Energy Consumption`) #As a check
```

```
## [1] "numeric"
```

```r
#new data frame of interest
converted_df = cbind(date_of_interest, Solar_Wind)
head(converted_df, 5) #As a check
```

```
##   date_of_interest Solar Energy Consumption Wind Energy Consumption
## 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
```

```r
class(converted_df)#As a check
```

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

```r
#dropping NA
Solar_Wind = drop_na(Solar_Wind)
head(Solar_Wind, 5) #As a check
```

```
##   Solar Energy Consumption Wind Energy Consumption
## 1                   -0.001                   0.000
## 2                    0.001                   0.002
## 3                    0.002                   0.002
## 4                    0.003                   0.006
## 5                    0.007                   0.008
```

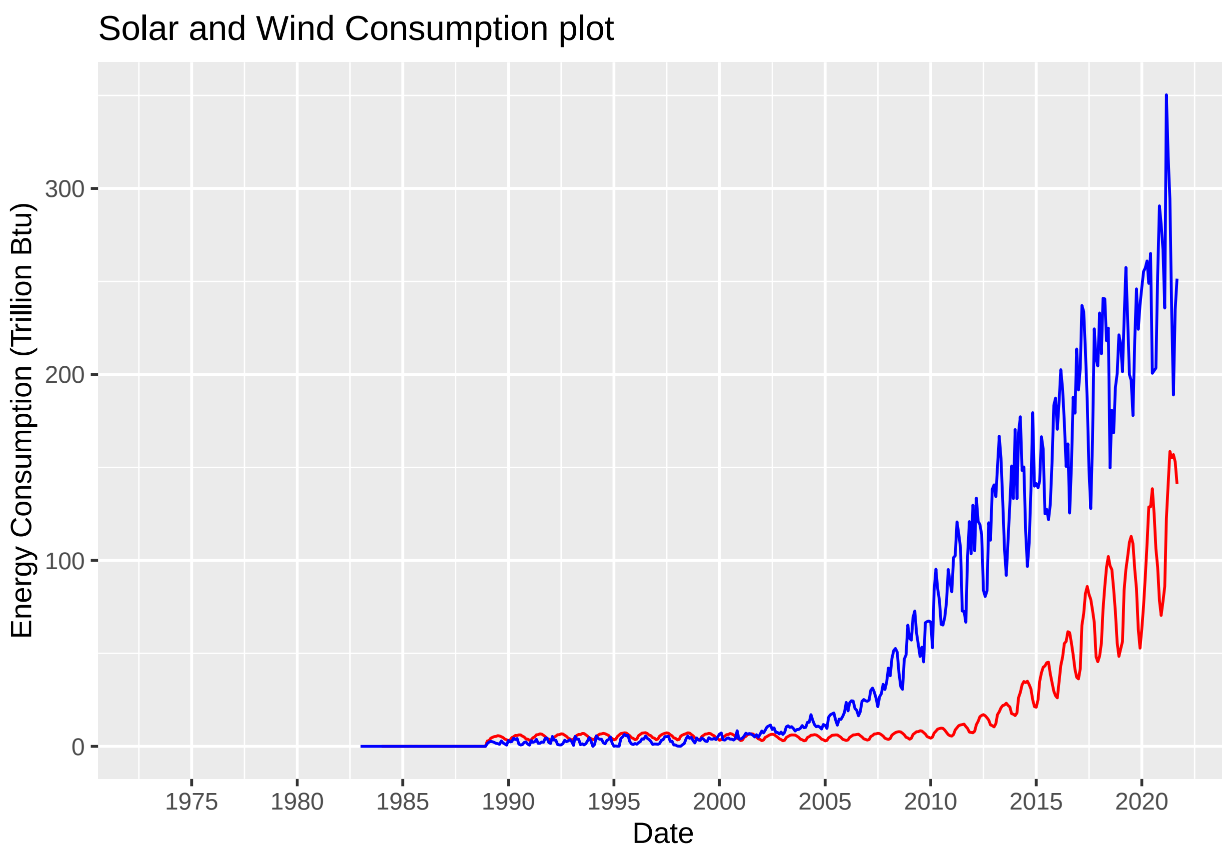
Q2

Plot the Solar and Wind energy consumption over time using ggplot. Plot each series on a separate graph. No need to add legend. Add informative names to the y axis using `ylab()`. 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")")*

```r
ggplot(data = converted_df) +
  geom_line(aes(x = converted_df[, 1], y = converted_df[, 2]), color = "Red") +
  geom_line(aes(x = converted_df[, 1], y = converted_df[, 3]), color = "Blue") +
  ylab("Energy Consumption (Trillion Btu)") +
  xlab("Date") +
  scale_x_date(date_breaks = "5 years", date_labels = "%Y") +
  ggtitle("Solar and Wind Consumption plot")
```

```
## Warning: Removed 132 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 120 row(s) containing missing values (geom_path).
```



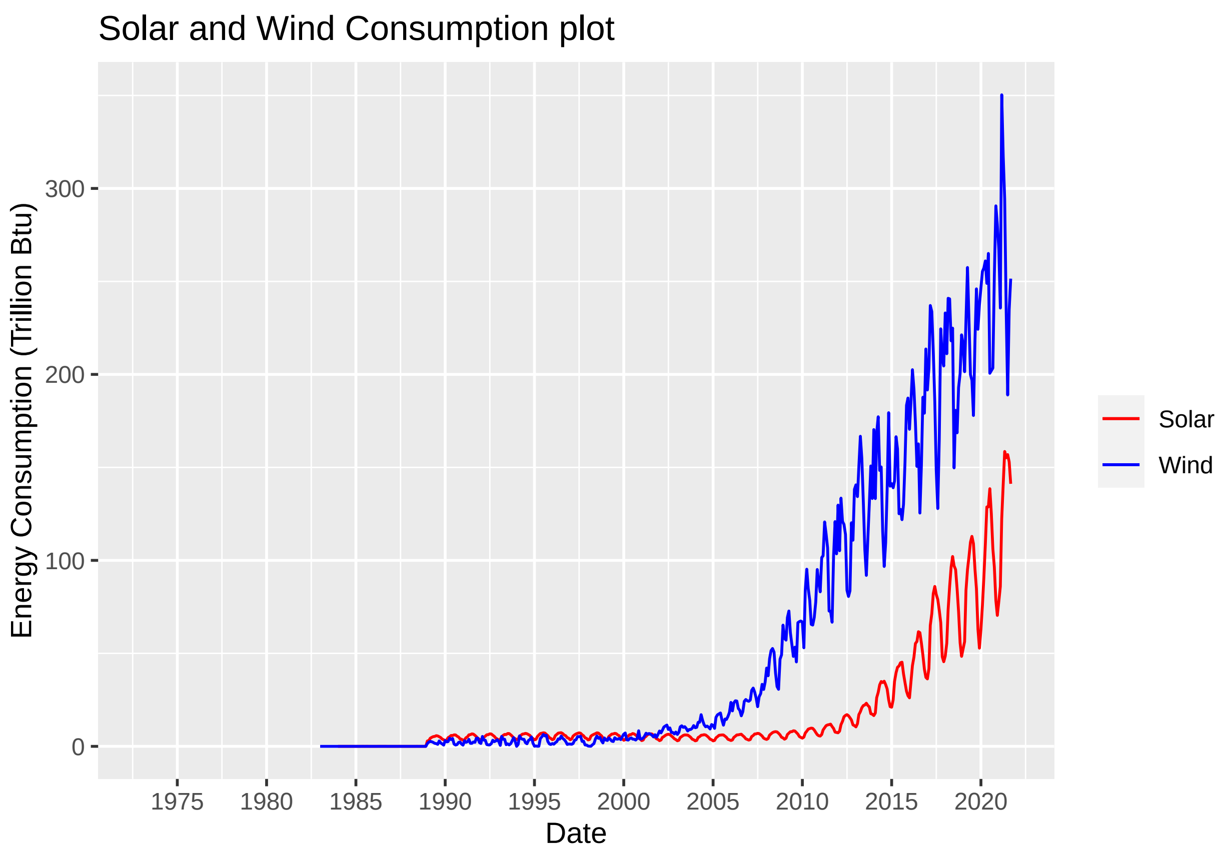
Q3

Now plot both series in the same graph, also using ggplot(). Look at lines 142-149 of the file `05_Lab_OutliersMissingData_Solution` to learn how to manually add a legend to ggplot. Make the solar energy consumption red and wind energy consumption blue. Add informative name to the y axis using `ylab("Energy Consumption)`. And use function scale_x_date() again to improve x axis.

```
ggplot(data = converted_df) +
  geom_line(aes(x = converted_df[, 1], y = converted_df[, 2], color = "Solar")) +
  geom_line(aes(x = converted_df[, 1], y = converted_df[, 3], color = "Wind")) +
  ylab("Energy Consumption (Trillion Btu)") +
  xlab("Date") +
  scale_colour_manual("",
                      breaks = c("Solar", "Wind"),
                      values = c("Solar"="red", "Wind"="blue")) +
  scale_x_date(date_breaks = "5 years", date_labels = "%Y") +
  ggtitle("Solar and Wind Consumption plot")
```

```
## Warning: Removed 132 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 120 row(s) containing missing values (geom_path).
```



Q3

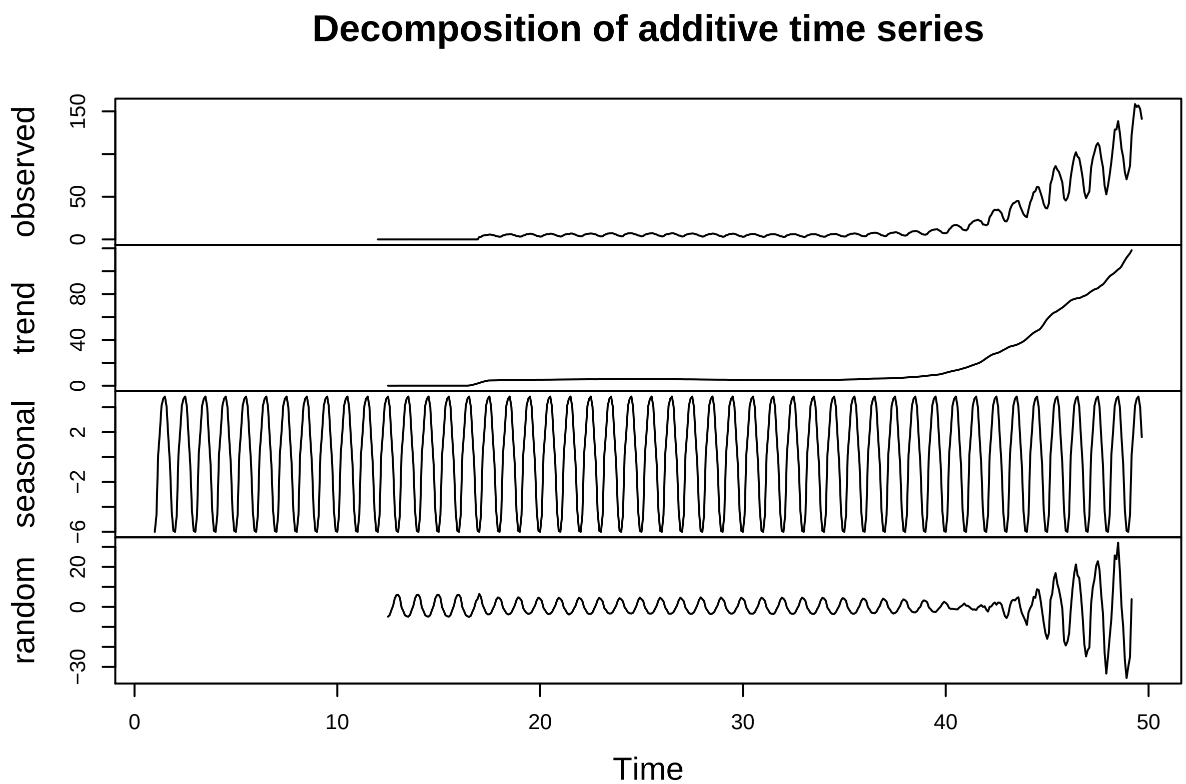
Transform wind and solar series into a time series object and apply the decompose function on them using the additive option, i.e., `decompose(ts_data, type = "additive")`. What can you say about the trend component? What about the random component? Does the random component look random? Or does it appear to still have some seasonality on it?

```
#converting to time series object
converted_df_ts = ts(converted_df, frequency = 12)
class(converted_df_ts) #As a check
```

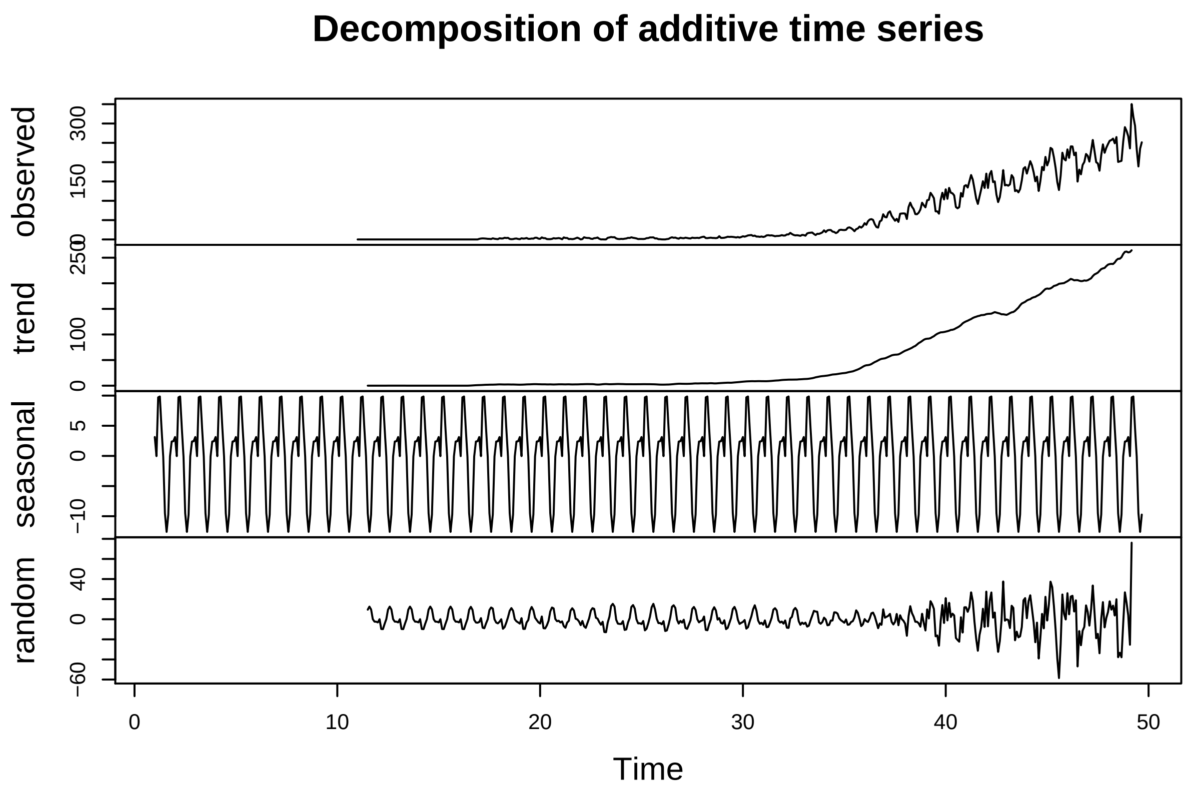
```
## [1] "mts"    "ts"     "matrix"
```

```
#Decomposing
Solar_decompose = decompose(converted_df_ts[, 2], type = "additive")
Wind_decompose = decompose(converted_df_ts[, 3], type = "additive")

#plotting the decomposed data
plot(Solar_decompose)
```




```
plot(Wind_decompose)
```



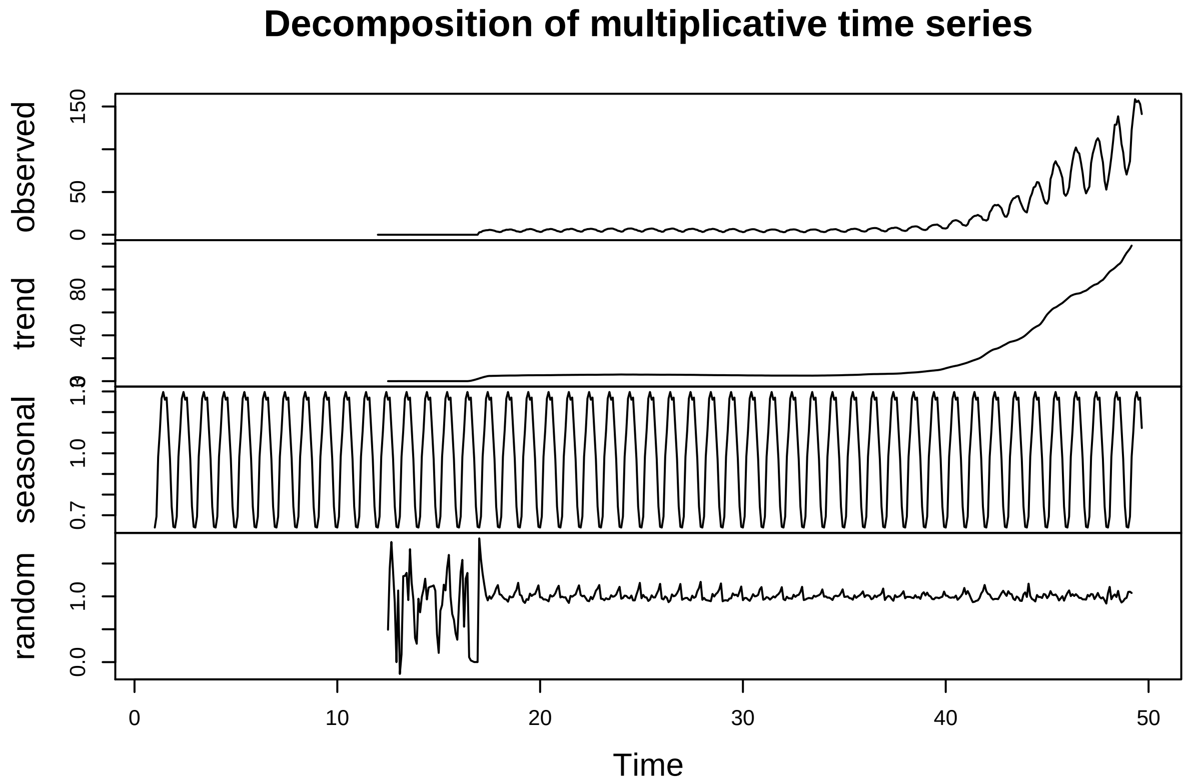
The trend appears to be increasing for both solar and wind consumption - positive trend. The random component has seasonality in it for both solar and wind consumption although, its more conspicuous for solar than wind.

Q4

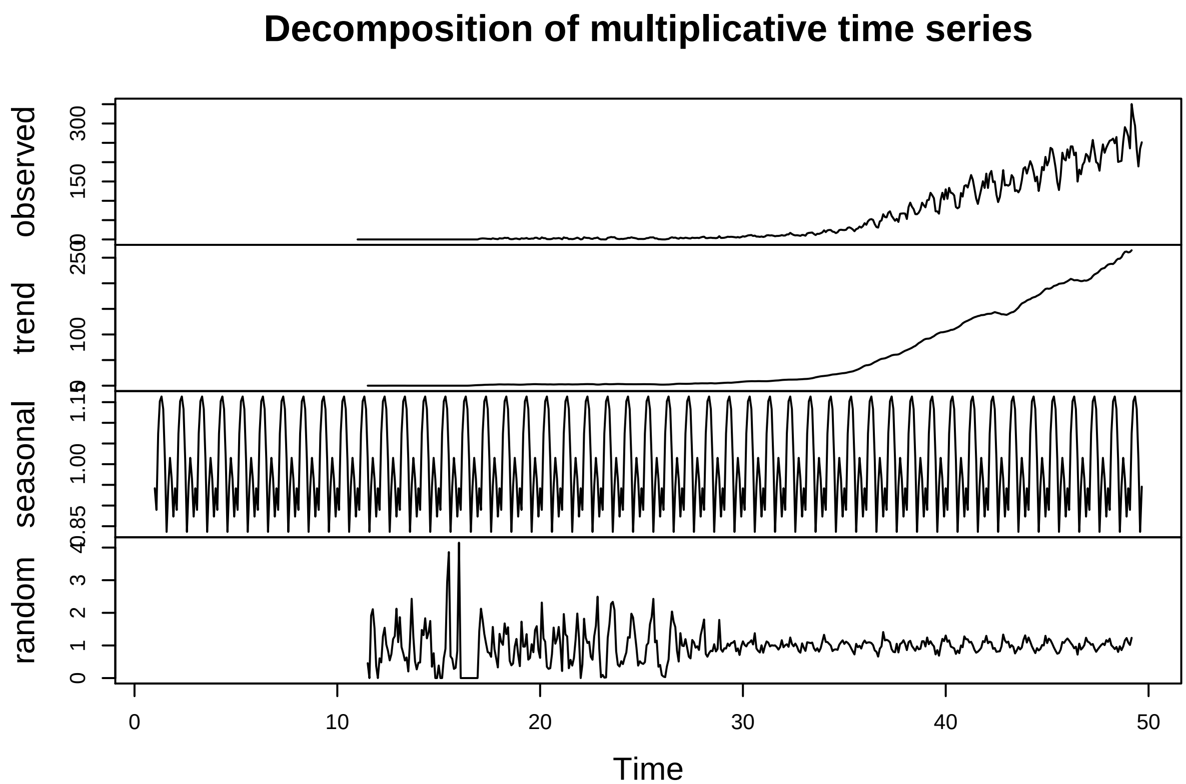
Use the decompose function again but now change the type of the seasonal component from additive to multiplicative. What happened to the random component this time?

```
#Decomposing
Solar_decompose = decompose(converted_df_ts[, 2], type = "multiplicative")
Wind_decompose = decompose(converted_df_ts[, 3], type = "multiplicative")

#plotting the decomposed data
plot(Solar_decompose)
```




```
plot(Wind_decompose)
```



The random component has become a lot better in that the seasonality has been removed, although not completely, due to the introduction of type="multiplicative"for both solar and wind consumption.

Q5

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

> Answer: Well, it doesn't look like we need all of the histroical data from 90s. It looks like we can just start from 2000 for the wind power consumption and start from 2010 for the solar power consumption. the reason for this is because, prior to those dates, the trend polts for both of 'em remains constant and remain closer to 0. Therefore, any further data prior won't help in better fitting the model and forecasting future values.

Q6

Create a new time series object where historical data starts on January 2012. Hint: use `filter()` function so that you don't need to point to row numbers, .i.e, `filter(xxxx, year(Date) >= 2012 )`. Apply the decompose function `type=additive` to this new time series. Comment the results. Does the random component look random? Think about our discussion in class about trying to remove the seasonal component and the challenge of trend on the seasonal component.

```r
new_converted_df = converted_df %>%
  filter(year(converted_df[, 1]) >= 2012)

head(new_converted_df, 50) #As a check
```

```
##    date_of_interest Solar Energy Consumption Wind Energy Consumption
## 1        2012-01-01                    7.288                 129.726
## 2        2012-02-01                    8.165                 105.171
## 3        2012-03-01                   11.678                 133.476
## 4        2012-04-01                   13.478                 120.941
## 5        2012-05-01                   15.933                 119.336
## 6        2012-06-01                   16.651                 113.928
## 7        2012-07-01                   17.063                  83.946
## 8        2012-08-01                   16.478                  80.590
## 9        2012-09-01                   15.384                  83.642
## 10       2012-10-01                   14.153                 120.241
## 11       2012-11-01                   11.547                 110.848
## 12       2012-12-01                   11.142                 138.215
## 13       2013-01-01                   10.449                 140.620
## 14       2013-02-01                   12.205                 134.295
## 15       2013-03-01                   17.039                 150.325
## 16       2013-04-01                   18.627                 166.741
## 17       2013-05-01                   20.722                 154.933
## 18       2013-06-01                   21.970                 131.171
## 19       2013-07-01                   22.251                 105.844
## 20       2013-08-01                   23.202                  91.917
## 21       2013-09-01                   21.998                 111.382
## 22       2013-10-01                   21.221                 130.092
## 23       2013-11-01                   17.543                 150.779
## 24       2013-12-01                   17.297                 133.260
## 25       2014-01-01                   16.542                 170.336
## 26       2014-02-01                   17.939                 133.222
## 27       2014-03-01                   26.187                 168.668
## 28       2014-04-01                   29.029                 177.224
## 29       2014-05-01                   33.086                 148.369
## 30       2014-06-01                   34.867                 150.247
## 31       2014-07-01                   34.299                 115.902
## 32       2014-08-01                   35.031                  96.722
## 33       2014-09-01                   33.184                 109.553
## 34       2014-10-01                   30.860                 137.970
## 35       2014-11-01                   25.046                 179.424
## 36       2014-12-01                   21.350                 139.904
## 37       2015-01-01                   21.048                 141.296
## 38       2015-02-01                   25.062                 139.054
## 39       2015-03-01                   34.925                 142.655
## 40       2015-04-01                   39.575                 166.504
## 41       2015-05-01                   42.535                 159.833
## 42       2015-06-01                   43.229                 125.073
## 43       2015-07-01                   44.959                 127.442
## 44       2015-08-01                   45.266                 121.893
## 45       2015-09-01                   38.976                 130.201
## 46       2015-10-01                   34.293                 152.646
## 47       2015-11-01                   29.646                 183.414
```

```
## 48       2015-12-01                    27.219                 187.297
## 49       2016-01-01                    26.077                 170.482
## 50       2016-02-01                    35.136                 185.916
```

```
#converting to time series object
converted_df_new_ts = ts(new_converted_df, frequency = 12)
head(converted_df_new_ts, 5) #As a check
```

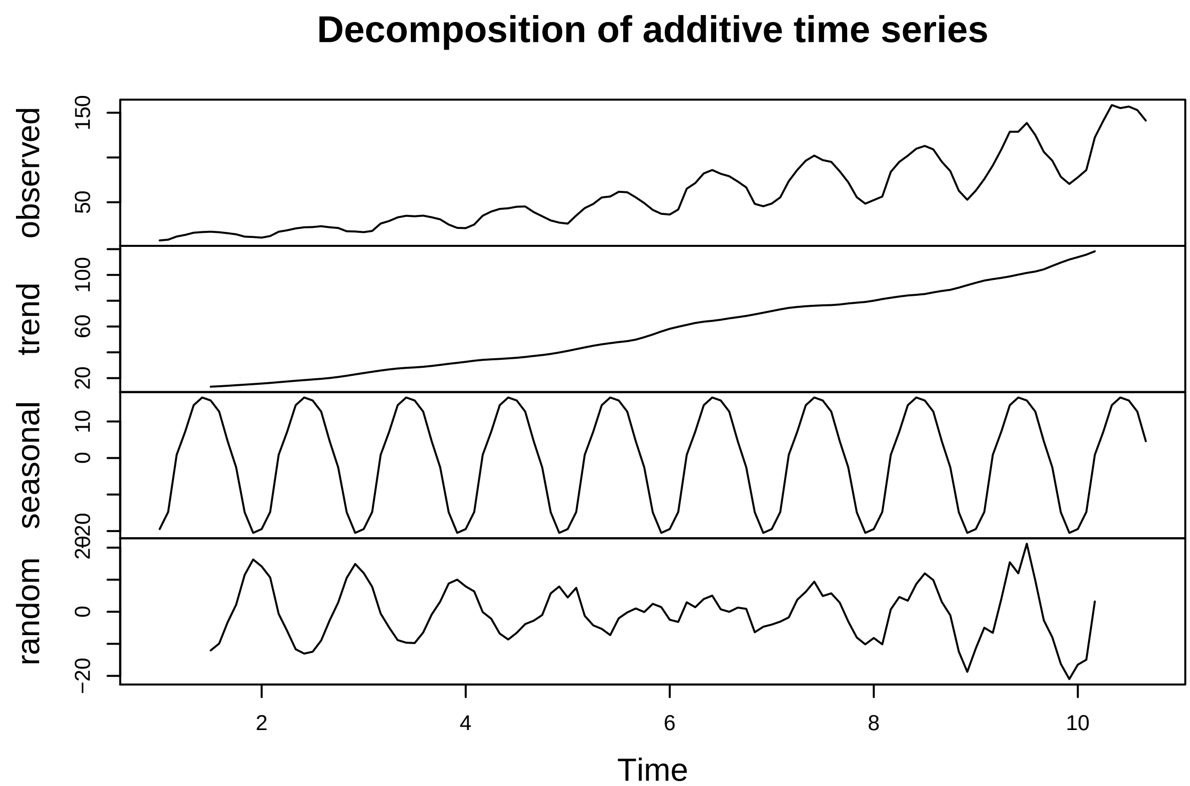
```
##        date_of_interest Solar Energy Consumption Wind Energy Consumption
## Jan 1             15340                    7.288                 129.726
## Feb 1             15371                    8.165                 105.171
## Mar 1             15400                   11.678                 133.476
## Apr 1             15431                   13.478                 120.941
## May 1             15461                   15.933                 119.336
```

```
class(converted_df_new_ts) #As a check
```

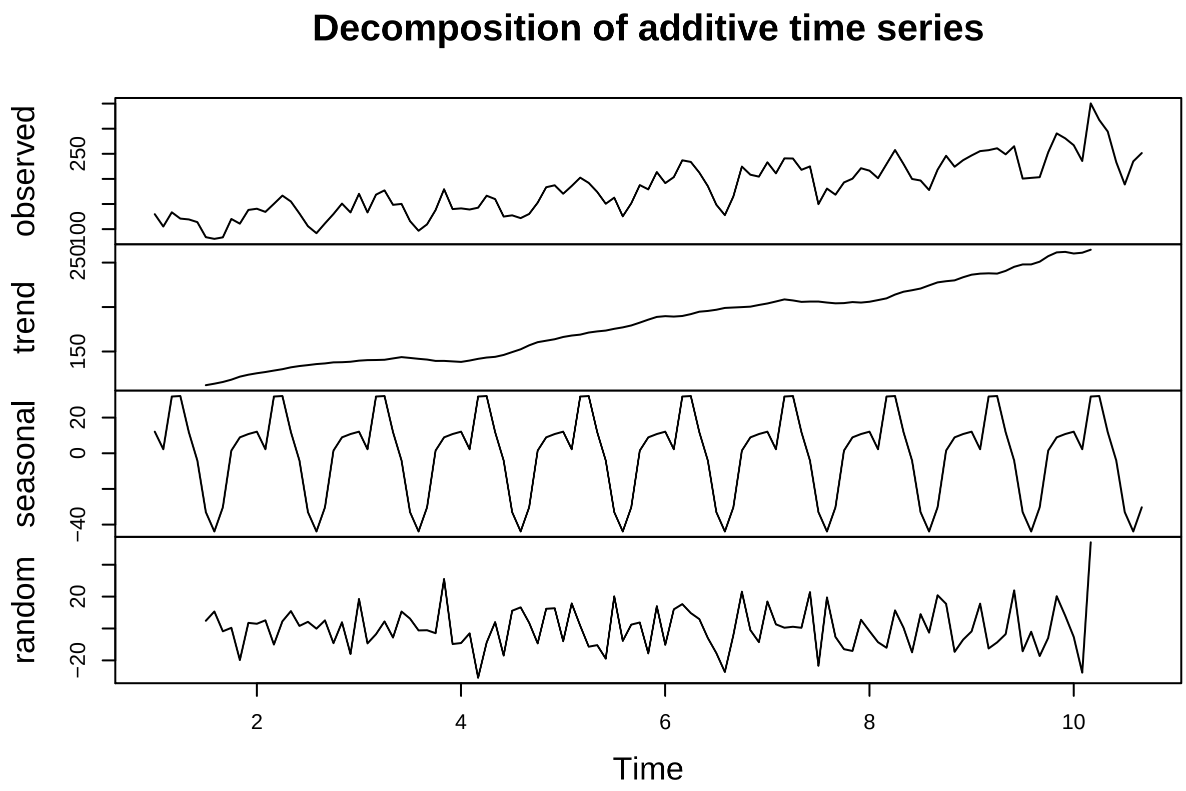
```
## [1] "mts"    "ts"     "matrix"
```

```
#Decomposing
Solar_new_decompose = decompose(converted_df_new_ts[, 2], type = "additive")
Wind_new_decompose = decompose(converted_df_new_ts[, 3], type = "additive")

#plotting the decomposed data
plot(Solar_new_decompose)
```



```
plot(Wind_new_decompose)
```



Answer: There is a massive shift now. First. the trend, as before is indeed increasing but, this time we see a much steady increase for both solar and wind consumption. With regard to random component, there is seasonality in both the solar consumption and wind consumption but it is very apparent in solar than it is for wind. Differencing is one possible way to remove seasonlity at this point.