ENV 790.30 - Time Series Analysis for Energy Data | Spring 2023

Assignment 5 - Due date 02/27/23

Hugh Cipparone

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_A05_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(forecast)
```

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

```
library(tseries)
library(ggplot2)
library(Kendall)
library(lubridate)
```

```
## Loading required package: timechange

##
## Attaching package: 'lubridate'

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

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

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
```

```
## v purrr   0.3.5
## -- 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()
```

```
library(cowplot)
```

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

Decomposing Time Series

Consider the same data you used for A04 from the spreadsheet "Table_10.1_Renewable_Energy_Production_and_Consumptio

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

head(energy_data)
```

```
##           Month Wood.Energy.Production Biofuels.Production
## 1  1973 January                129.630       Not Available
## 2 1973 February                117.194       Not Available
## 3    1973 March                129.763       Not Available
## 4    1973 April                125.462       Not Available
## 5      1973 May                129.624       Not Available
## 6     1973 June                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!

```
energy_data_edit<-energy_data %>%
  select(Solar.Energy.Consumption, Wind.Energy.Consumption,Month) %>%
  mutate_at(c("Solar.Energy.Consumption", "Wind.Energy.Consumption"), as.numeric) %>%
  drop_na()
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

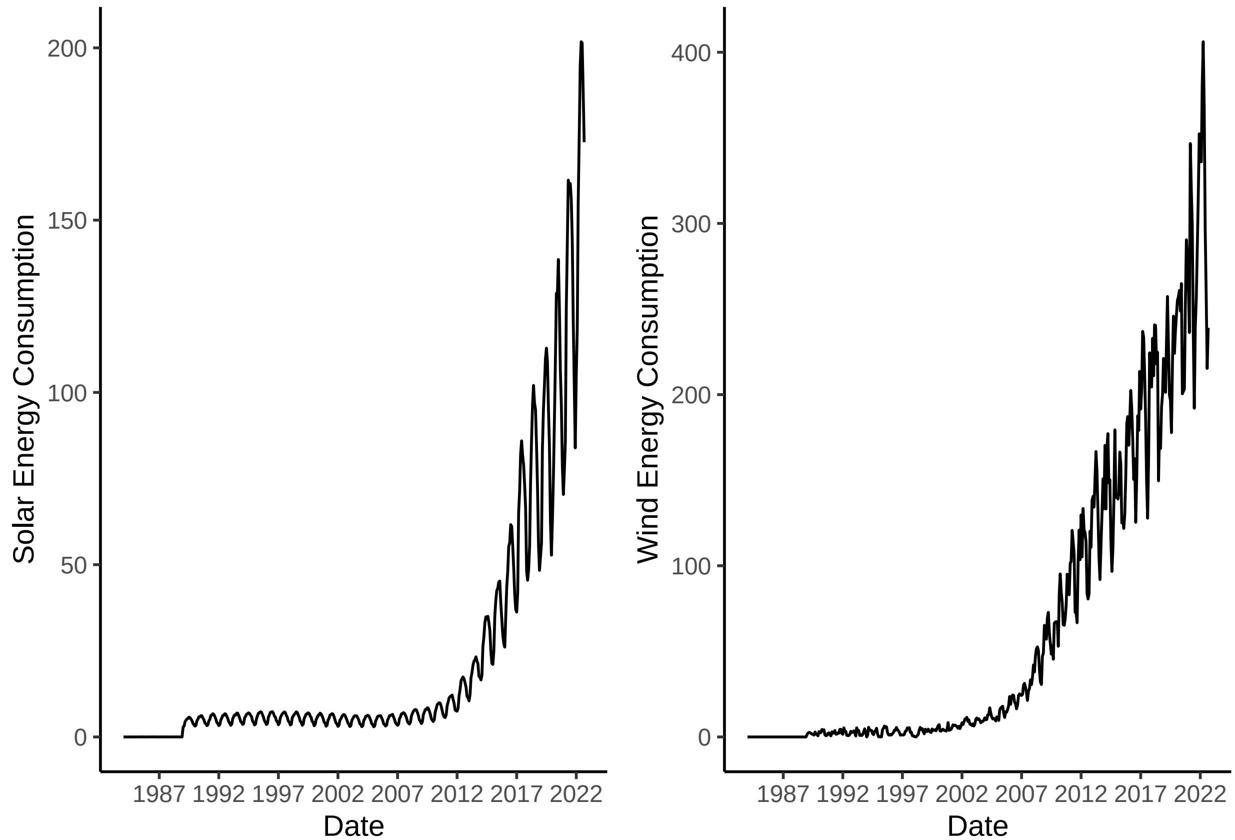
```
energy.date<-ym(energy_data_edit$Month)

energy_data_cleaned<-cbind(energy_data_edit,energy.date)%>%
  select(-Month)
```

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")")*

```
solar<-ggplot(energy_data_cleaned, aes(x=energy.date, y=Solar.Energy.Consumption))+
         geom_line()+
        theme_classic()+
        ylab("Solar Energy Consumption")+
        scale_x_date(date_breaks = "5 years", date_labels = "%Y")+
        xlab("Date")
```

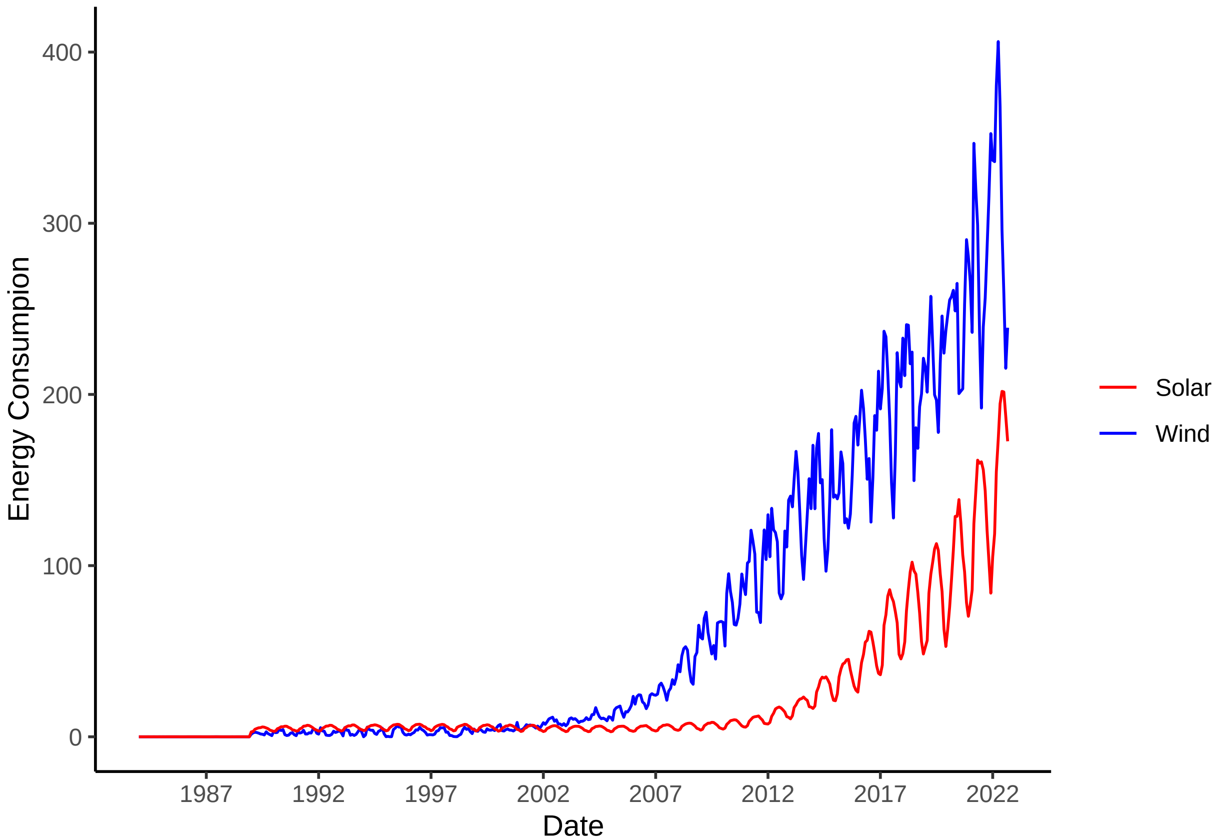
```
wind<-ggplot(energy_data_cleaned, aes(x=energy.date, y=Wind.Energy.Consumption))+
         geom_line()+
        theme_classic()+
  ylab("Wind Energy Consumption")+
  scale_x_date(date_breaks = "5 years", date_labels = "%Y")+
  xlab("Date")

plot_grid(solar,wind)
```



Q3

Now plot both series in the same graph, also using ggplot(). Look at lines 141-148 of the file `M4_OutliersMissingData_Part2_Complete.Rmd` 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(energy_data_cleaned)+
  geom_line(aes(x=energy.date, y=Wind.Energy.Consumption, color = "Wind"))+
  geom_line(aes(x=energy.date, y=Solar.Energy.Consumption, color = "Solar"))+
  labs(color="")+
  scale_color_manual(values = c("Wind" = "blue", "Solar" = "red")) +
  theme(legend.position = "bottom") +
  ylab(label="Energy Consumpion") +
  theme_classic()+
  scale_x_date(date_breaks = "5 years", date_labels = "%Y")+
  xlab("Date")
```

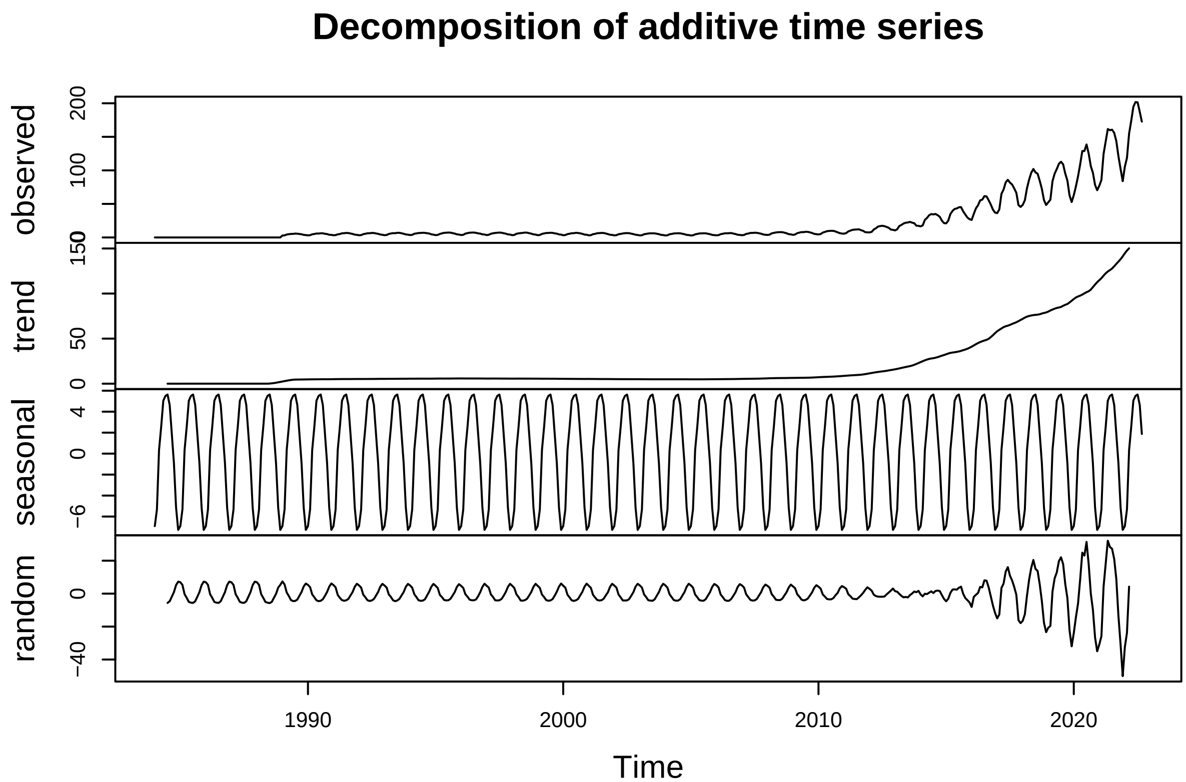


Q3

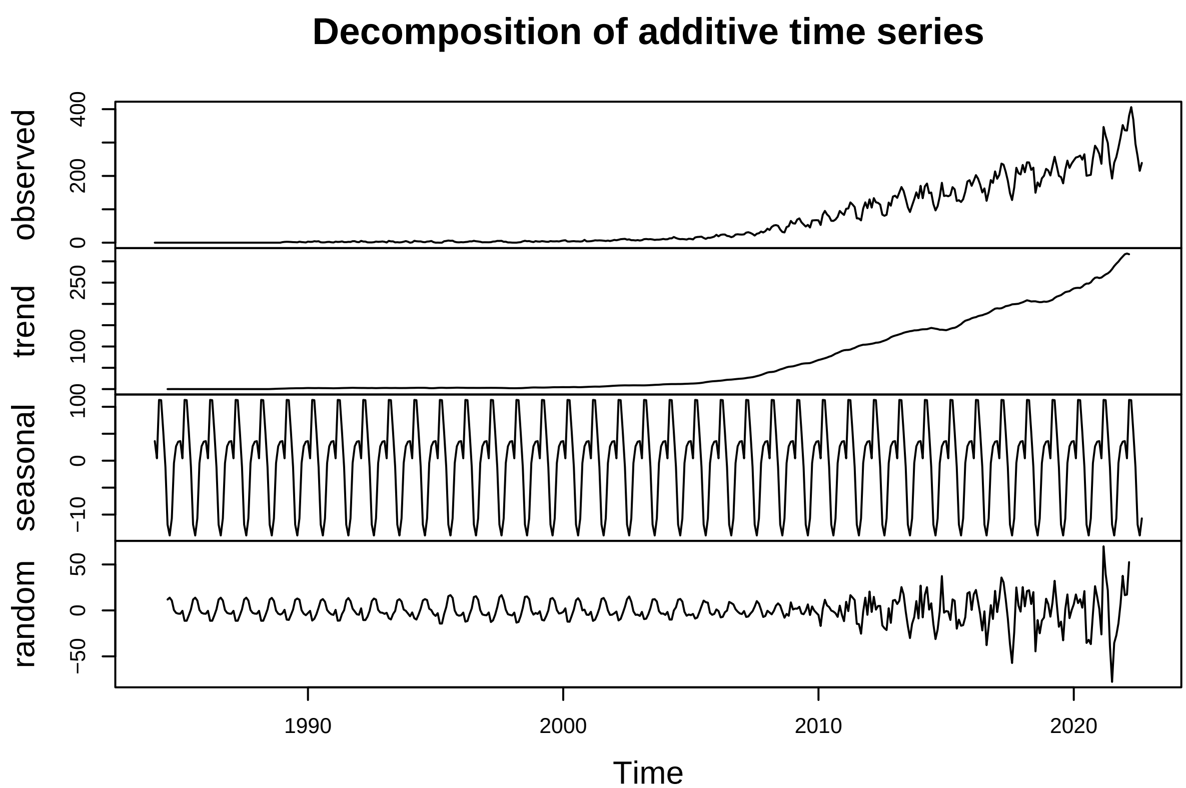
Transform wind and solar series into a time series object and apply the decompose function on them using the additive option, i.e., `decompose(ts_data, type = "additive")`. What can you say about the trend component? What about the random component? Does the random component look random? Or does it appear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

Answer: For solar, the trend is stationary and then grows significantly - the random component appears seasonal but then displays a clear trend (ie increasing magnitude of seasonal variability) starting around 2015. For wind I can see largely the same pattern.

```
energy.ts.sol<-ts(energy_data_cleaned[,1], frequency = 12, start=c(1984,1))
energy.ts.wind<-ts(energy_data_cleaned[,2], frequency = 12, start=c(1984,1))


decompose.energy.sol=decompose(energy.ts.sol, type="additive")
plot(decompose.energy.sol)
```




```r
decompose.energy.wind=decompose(energy.ts.wind, type="additive")
plot(decompose.energy.wind)
```

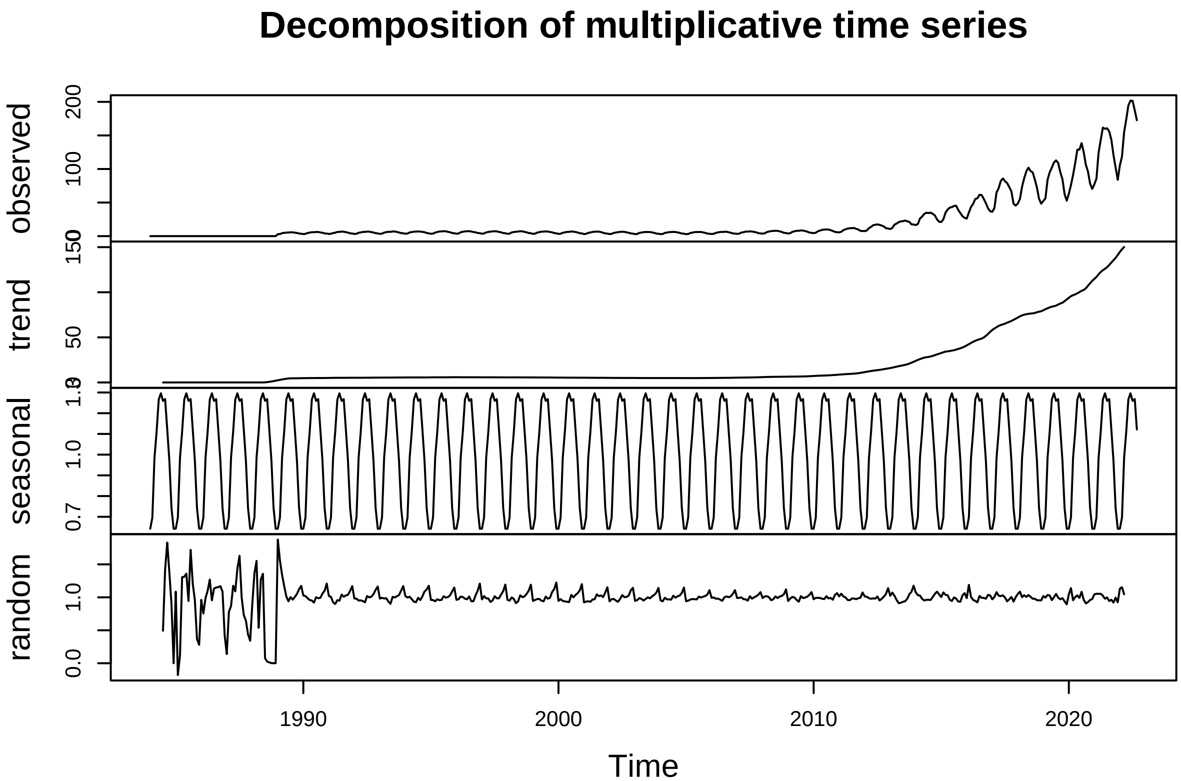


Q4

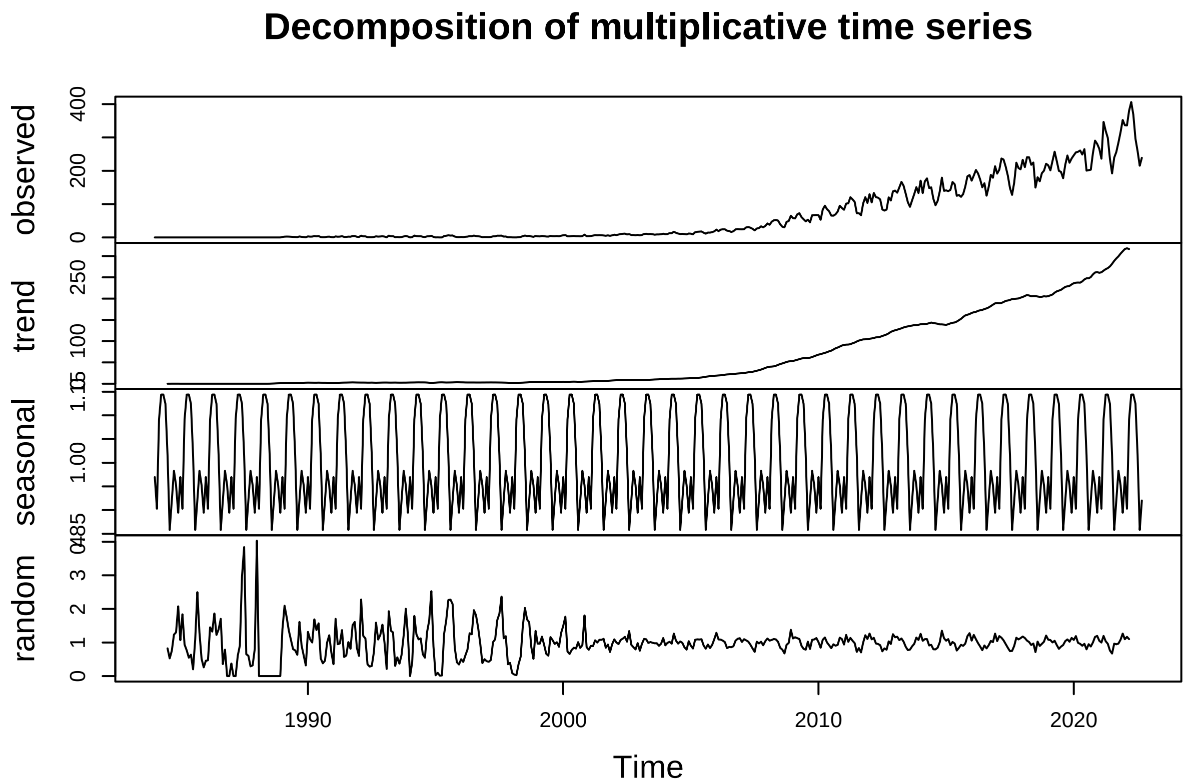
Use the decompose function again but now change the type of the seasonal component from additive to multiplicative. What happened to the random component this time?

Answer: The random component looks much more random!

```
decompose.energy.sol=decompose(energy.ts.sol, type="multiplicative")
plot(decompose.energy.sol)
```




```
decompose.energy.wind=decompose(energy.ts.wind, type="multiplicative")
plot(decompose.energy.wind)
```



Q5

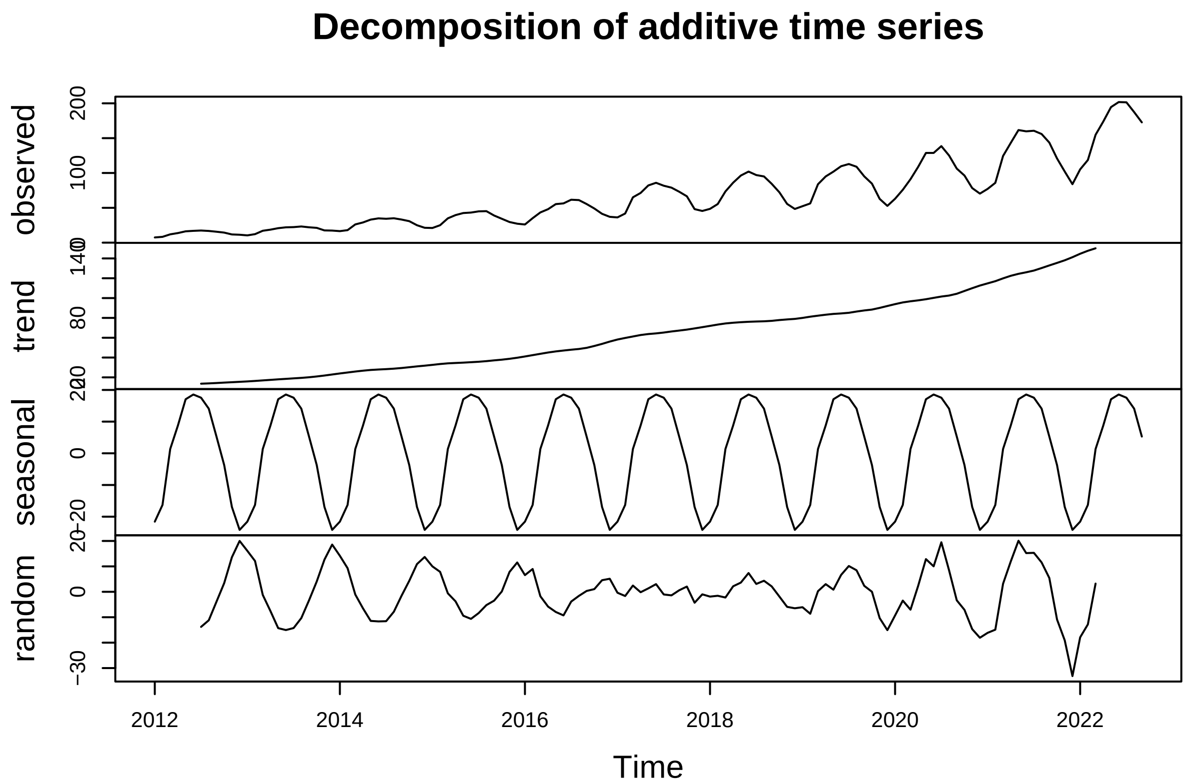
When fitting a model to this data, do you think you need all the historical data? Think about the data from 90s and early 20s. Are there any information from those years we might need to forecast the next six months of Solar and/or Wind consumption. Explain your response.

> Answer: Much of the components of this model seem relatively different between 1990 and 2000 and 2000 onward. In some ways this would suggest that you don't need that data. However, I do wonder what drove those early trends. Was there a social policy or condition that produced those trends that may return? If so that period might be valuable. In the absence of that, however, I'm not sure that you need those data.

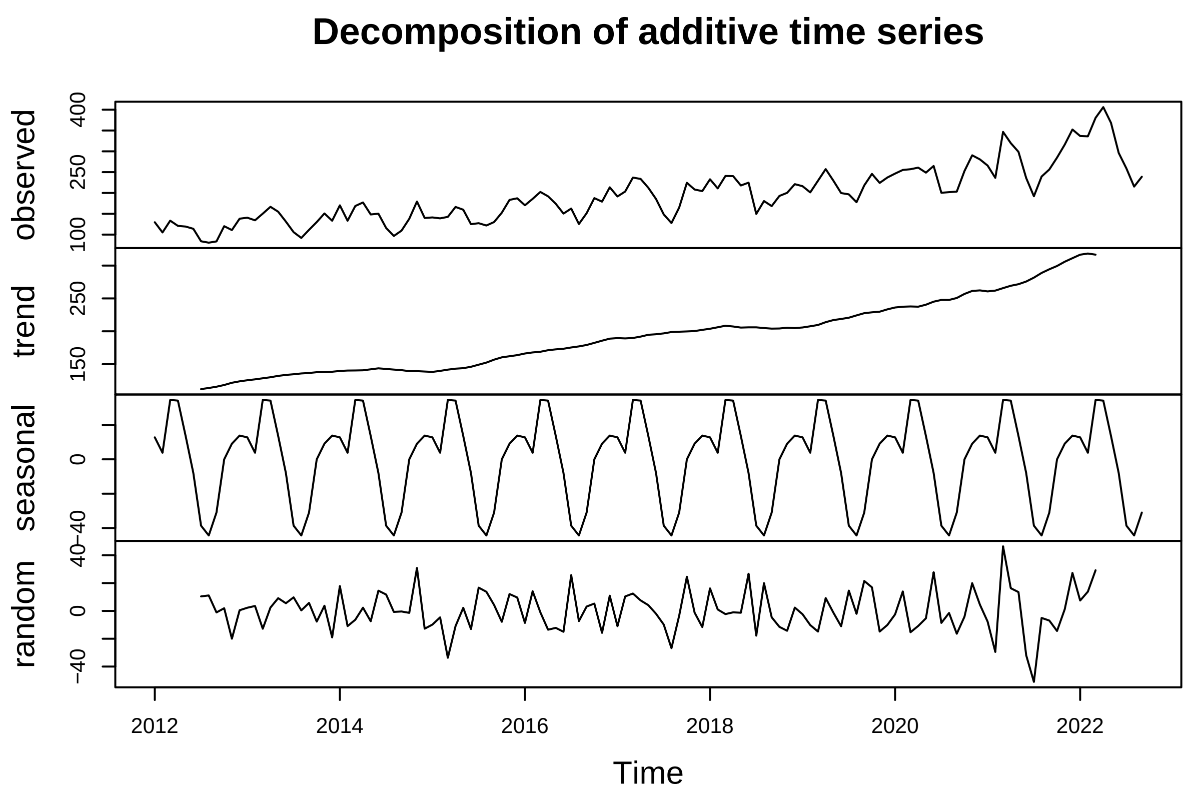
Q6

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 seasonal components that depends on the level of the series.

```
energy_data_cleaned_2012<-energy_data_cleaned %>%
  filter(year(energy.date) >= 2012)

energy.ts.sol.2012<-ts(energy_data_cleaned_2012[,1], frequency = 12, start=c(2012,1))
energy.ts.wind.2012<-ts(energy_data_cleaned_2012[,2], frequency = 12, start=c(2012,1))

decompose.energy.sol.2012=decompose(energy.ts.sol.2012, type="additive")
plot(decompose.energy.sol.2012)
```




```
decompose.energy.wind.2012=decompose(energy.ts.wind.2012, type="additive")
plot(decompose.energy.wind.2012)
```



Answer: The random component does look much more random for both - although its important to note that the mean of the random data for the solar energy consumption does not appear to be 0 so there may be other issues with the solar model. But overall - successful!