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

Assignment 5 - Due date 02/27/23

Kelsey Husted

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(xlsx)
library(gridExtra)
require(scales)
library(readxl)
library(openxlsx)
library(forecast)
library(tseries)
library(ggplot2)
library(Kendall)
library(lubridate)
library(tidyverse)  #load this package so yon clean the data frame using pipes
```

Decomposing Time Series

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

```
energy_data <- read.csv("./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source-Edit.c
#Transform date column
Date <- ym(energy_data[,1])
energy_data <- cbind(Date, energy_data[,2:10])

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 <- energy_data %>%
  select("Date", "Solar.Energy.Consumption", "Wind.Energy.Consumption") %>%
  subset(Solar.Energy.Consumption!= "Not Available" & Wind.Energy.Consumption!= "Not Available" )

energy_data$Solar.Energy.Consumption = as.numeric(energy_data$Solar.Energy.Consumption)
energy_data$Wind.Energy.Consumption = as.numeric(energy_data$Wind.Energy.Consumption)
head(energy_data)
```

```
##           Date Solar.Energy.Consumption Wind.Energy.Consumption
## 133 1984-01-01                   -0.001                   0.000
## 134 1984-02-01                    0.001                   0.002
## 135 1984-03-01                    0.002                   0.002
## 136 1984-04-01                    0.003                   0.006
## 137 1984-05-01                    0.007                   0.008
## 138 1984-06-01                    0.010                   0.006
```

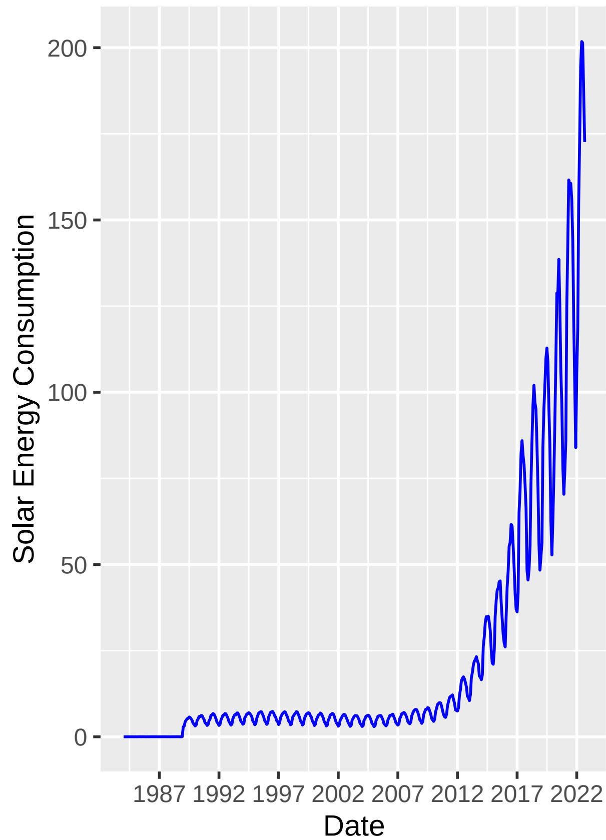
Q2

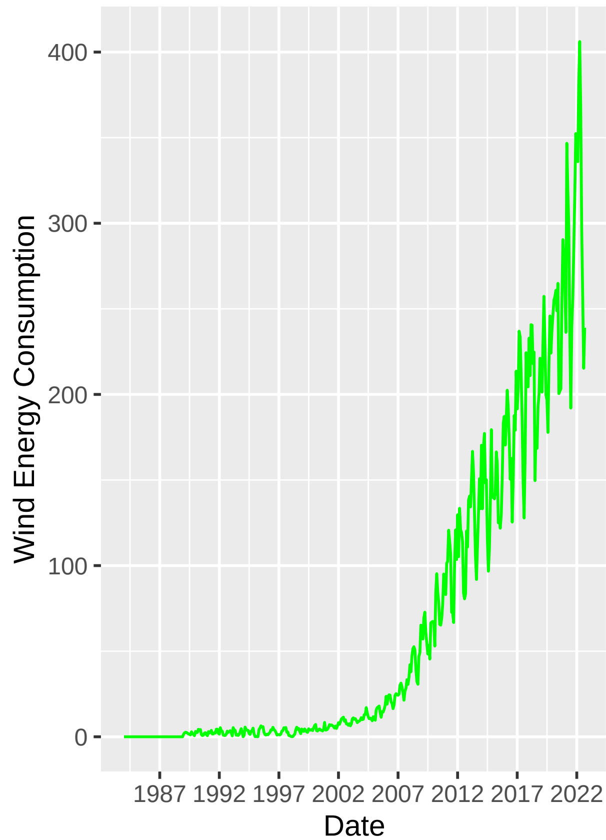
Plot the Solar and Wind energy consumption over time using ggplot. Plot each series on a separate graph. No need to add legend. Add informative names to the y axis using `ylab()`. Explore the function scale_x_date() on ggplot and see if you can change the x axis to improve your plot. Hint: use *scale_x_date(date_breaks = "5 years", date_labels = "%Y")"*)

```
plot1 <- ggplot(energy_data, aes(x=Date, y=Solar.Energy.Consumption)) +
  geom_line(color="blue") +
  ylab("Solar Energy Consumption") +
  #labs(y = "Solar Energy Consumption") +
  scale_x_date(breaks = breaks_width("5 years"), date_labels = "%Y")

plot2 <- ggplot(energy_data, aes(x=Date, y=Wind.Energy.Consumption)) +
  geom_line(color="green") +
  ylab("Wind Energy Consumption") +
  #labs(y="Wind Energy Consumption") +
  scale_x_date(breaks = breaks_width("5 years"), date_labels = "%Y")

grid.arrange(plot1, plot2, ncol=2)
```



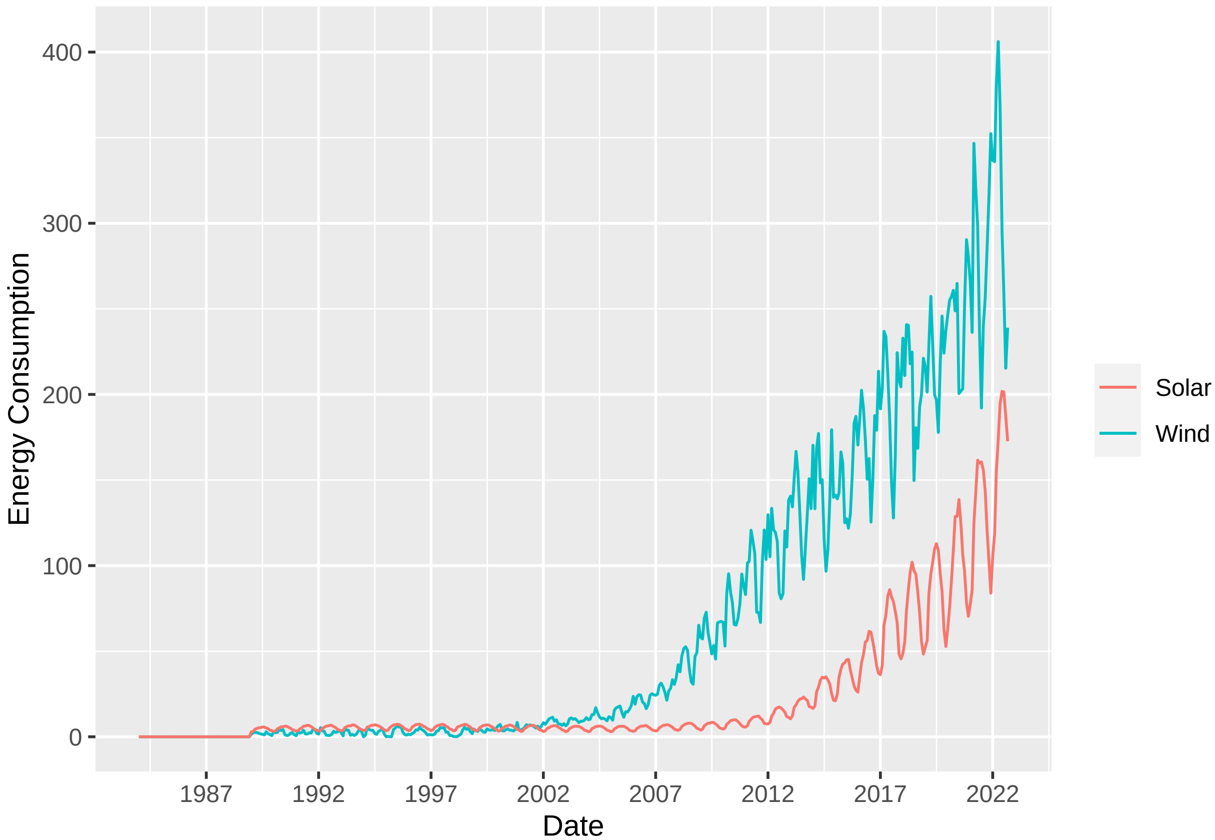


Q3

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, aes(x=Date, y=Wind.Energy.Consumption, colour = "Wind", col = "blue")) +
  geom_line() +
  geom_line(aes(y=Solar.Energy.Consumption, colour = "Solar", col = "red")) +
  #ylabs("Energy Consumption") +
  labs(y="Energy Consumption", color = " ") +
  scale_x_date(breaks = breaks_width("5 years"), date_labels = "%Y")
```

```
## Warning: Duplicated aesthetics after name standardisation: colour
## Duplicated aesthetics after name standardisation: colour
```



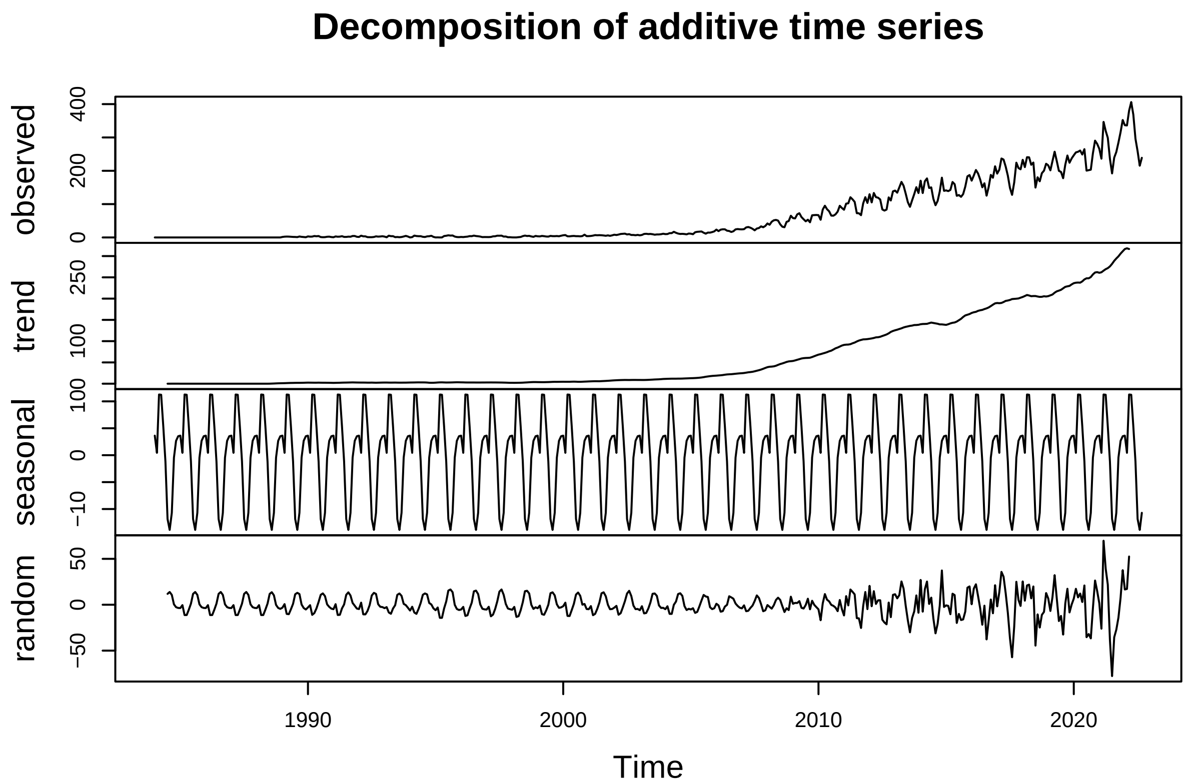

```
#used a different method to create a legend o I had to use labs(y=...) instead of the ylab(..) function
```

Q4

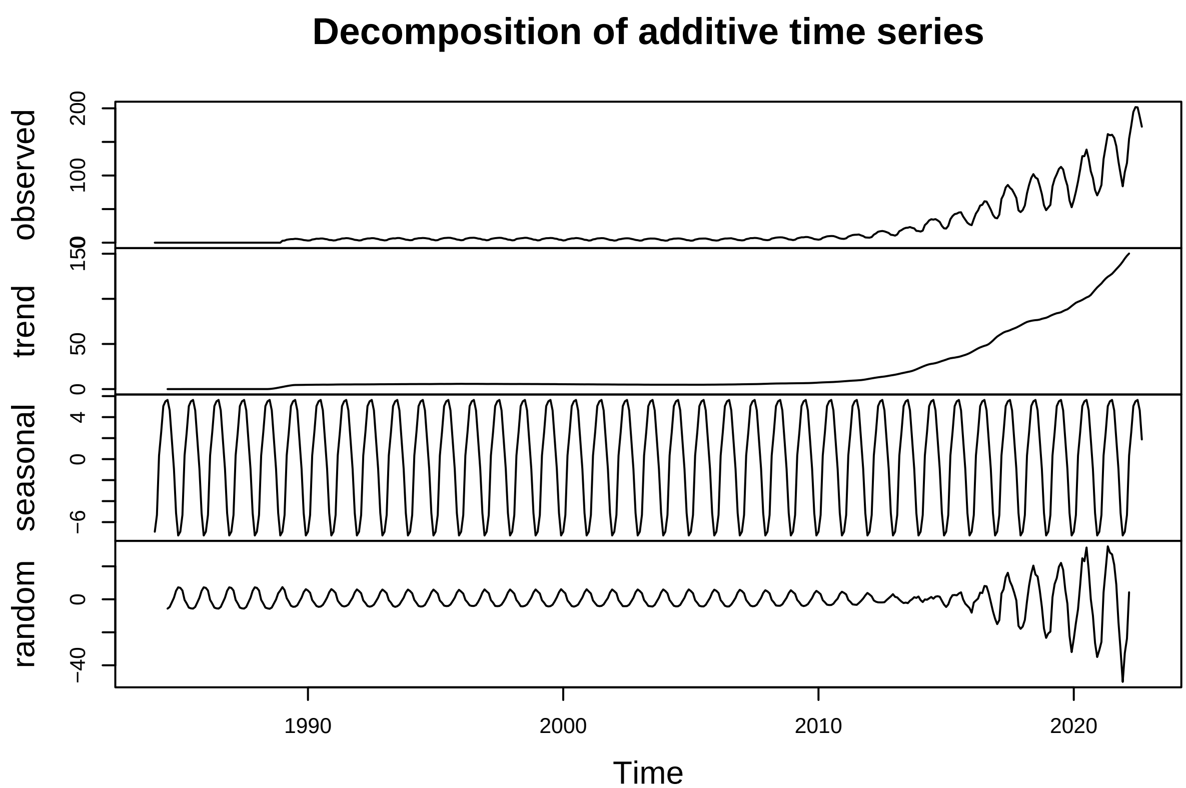
Transform wind and solar series into a time series object and apply the decompose function on them using the additive option, i.e., `decompose(ts_data, type = "additive")`. What can you say about the trend component? What about the random component? Does the random component look random? Or does it appear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

> Answer: For both solar and wind, the trend is noticeably increasing. Alternatively, the random components for solar and wind appear to be cyclic with a gradual increase in magnitude. Consequently, I think there may be some seasonality remaining in the radom component for both solar and wind data. The additive model is useful when seasonal variation is constant over time, so maybe seasonality increases over time instead.

```
#apply ts
ts_wind <- ts(energy_data[,3], frequency = 12, start = c(1984,1))
ts_solar <- ts(energy_data[,2], frequency = 12, start = c(1984,1))
#decompose with additive
decompose_wind_add <- decompose(ts_wind, type = "additive")
plot(decompose_wind_add)
```




```
decompose_solar_add <- decompose(ts_solar, type = "additive")
plot(decompose_solar_add)
```

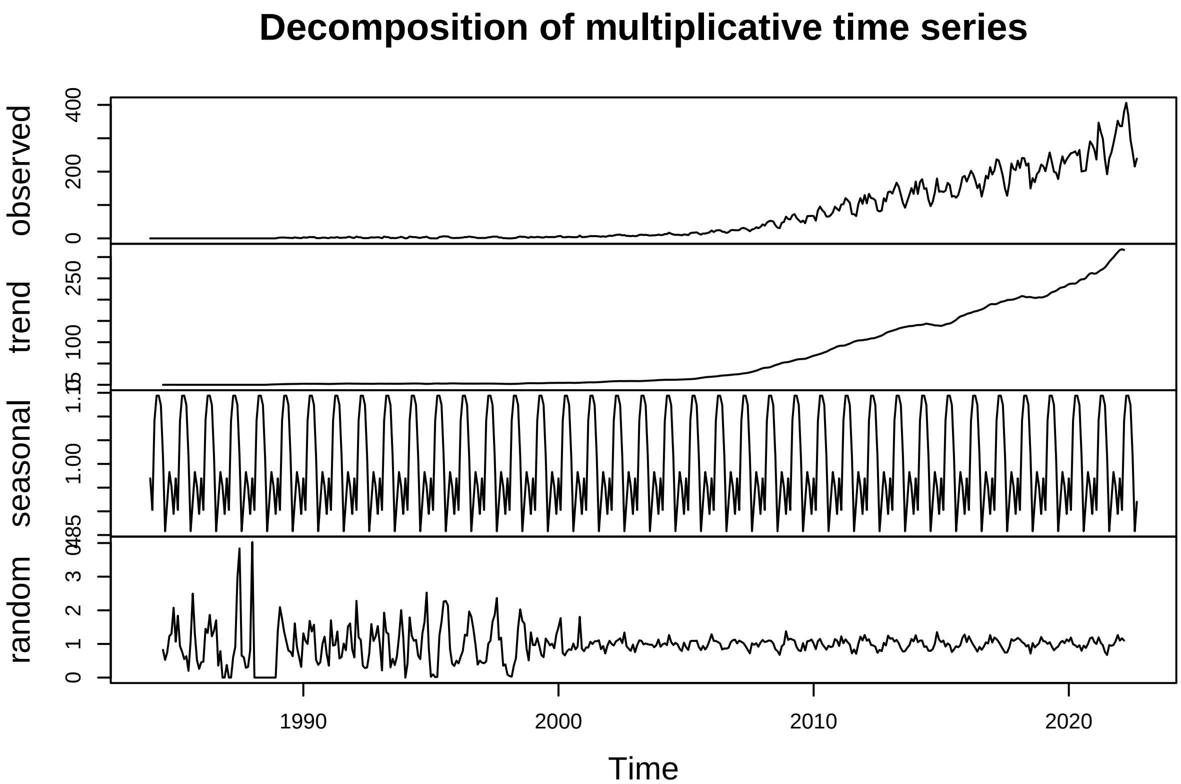


Q5

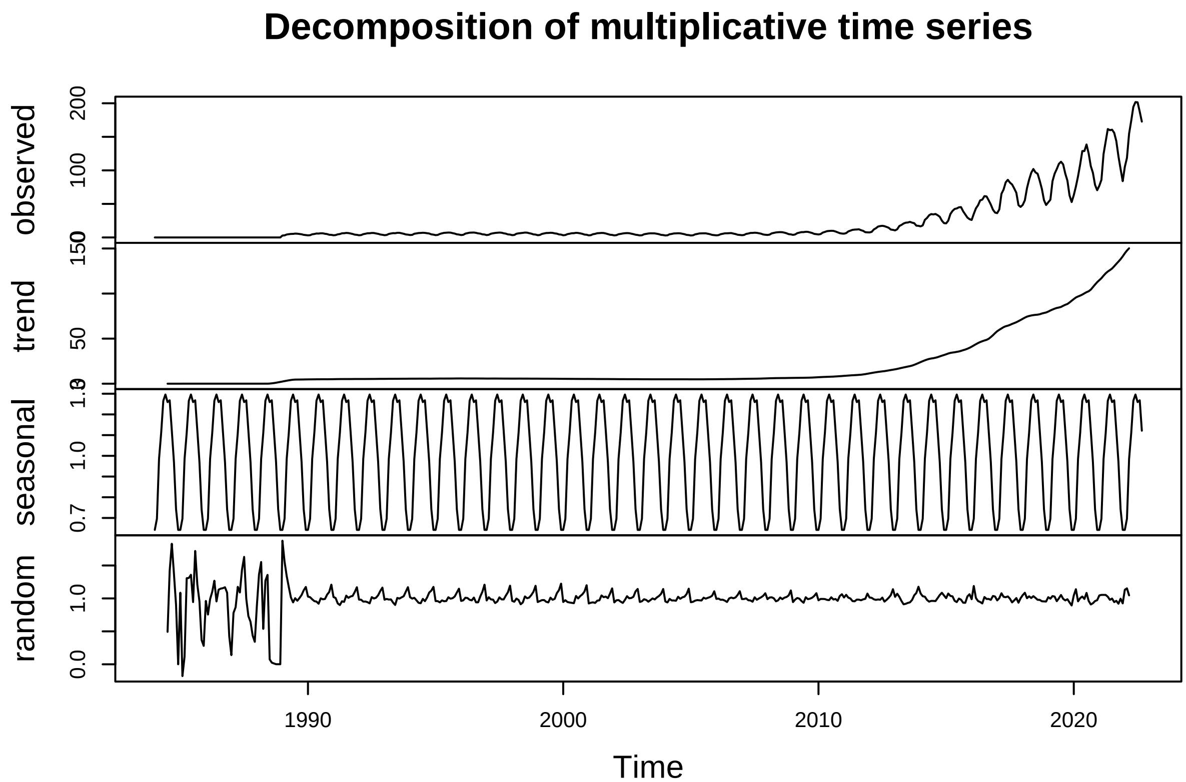
Use the decompose function again but now change the type of the seasonal component from additive to multiplicative. What happened to the random component this time?

> Anwer: It seems like the cyclic and additive nature of the random components were removed for both solar and wind consumption data. Futhermore, there isn't a gradual increase in cycles as seen in the decomposed plots of Q4.

```
#decompose with multiplicative
decompose_wind_mult <- decompose(ts_wind, type = "multiplicative")
plot(decompose_wind_mult)
```




```
decompose_solar_mult <- decompose(ts_solar, type = "multiplicative")
plot(decompose_solar_mult)
```



Q6

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.

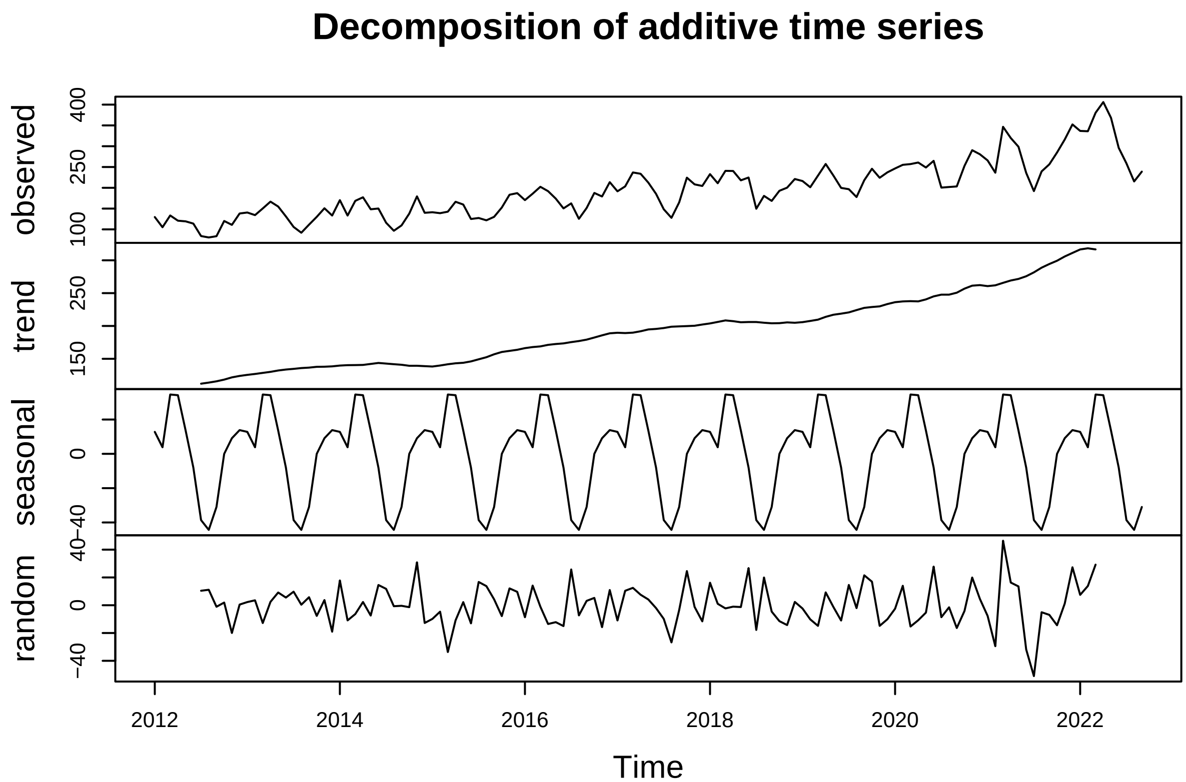
> Answer: I would assume that maybe having data over a longer period of time is generally better in practice with potentially higher statistical significance when forecasting over longer periods. Even so, without the historical data, one can still observe the upward trend and seasonality in the data and thus potentially forecast for the next six months. Maybe removing some historical data might help provide a more accurate forecast by omitting anomaly events.

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.

> Answer: The random components for both solar and wind consumption data appears to be random and very similar to the multiplicative decomposed random plot in Q5.

```r
#filter data
historical_data <- filter(energy_data, year(Date) >= 2012)
#run aother ts
ts_wind2 <- ts(historical_data[,3], frequency = 12, start = c(2012,1))
ts_solar2 <- ts(historical_data[,2], frequency = 12, start = c(2012,1))
#decompose
decompose_wind2 <- decompose(ts_wind2, type = "additive")
plot(decompose_wind2) #The decomposed wind plots display an upwards trend and seasonality.
```




```r
decompose_solar2 <- decompose(ts_solar2, type = "additive")
plot(decompose_solar2) #The decomposed solar plots display an upwards trend and seasonality as well.
```