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

Assignment 5 - Due date 02/28/22

Narissa Jimenez-Petchumrus

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.\

Decomposing Time Series

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

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!

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

#create data frame with date and 2 energy types
energy_data2 <- energy_data[, c(1,8:9)] %>% as.data.frame()
energy_data2
sapply(energy_data2, class)

#change column names
names(energy_data2) <- c('Month','Solar_Energy_Consumption','Wind_Energy_Consumption')
head(energy_data2)
sapply(energy_data2,class)

#convert to numeric and drop NAs
```

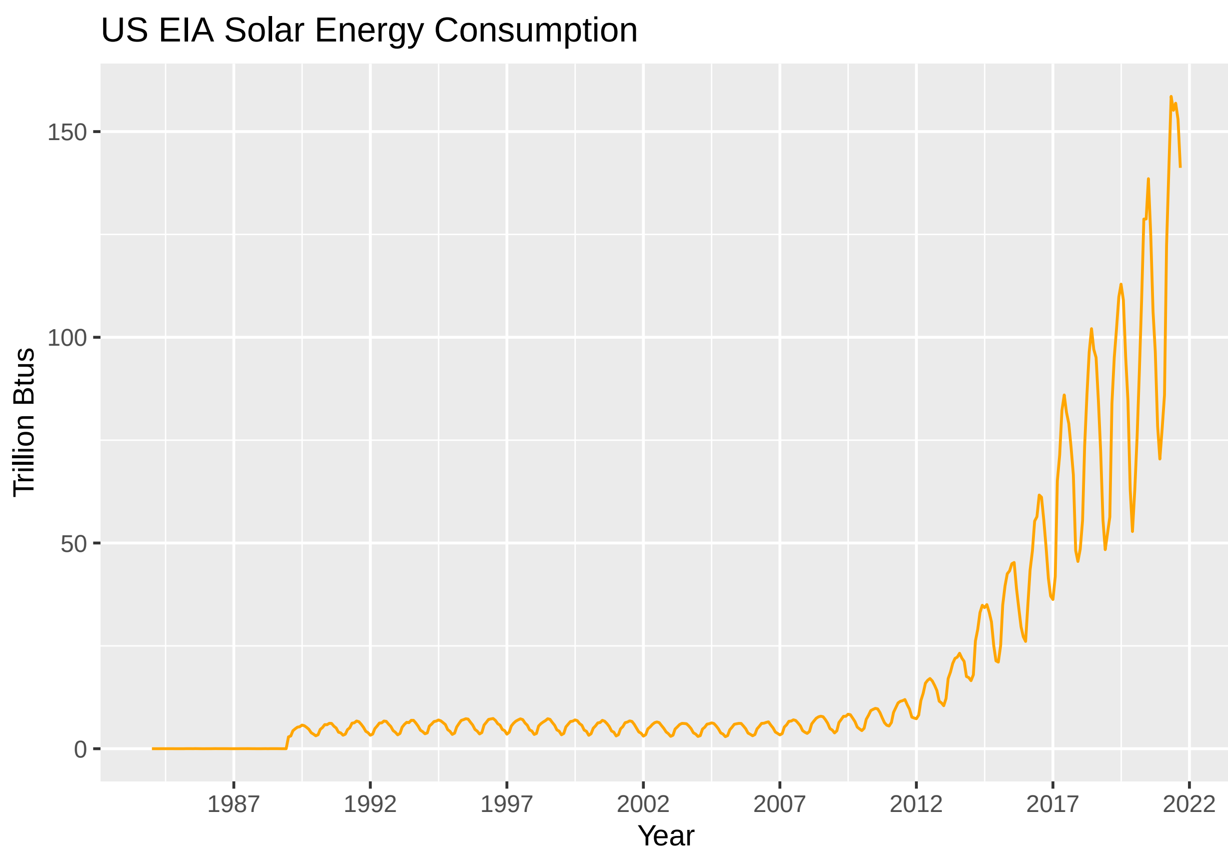
```
energy_data2$Solar_Energy_Consumption <- as.numeric(energy_data2$Solar_Energy_Consumption)
energy_data2$Wind_Energy_Consumption <- as.numeric(energy_data2$Wind_Energy_Consumption)
energy_data3 <- energy_data2 %>% drop_na()
energy_data3

sapply(energy_data3, class)
```

Q2

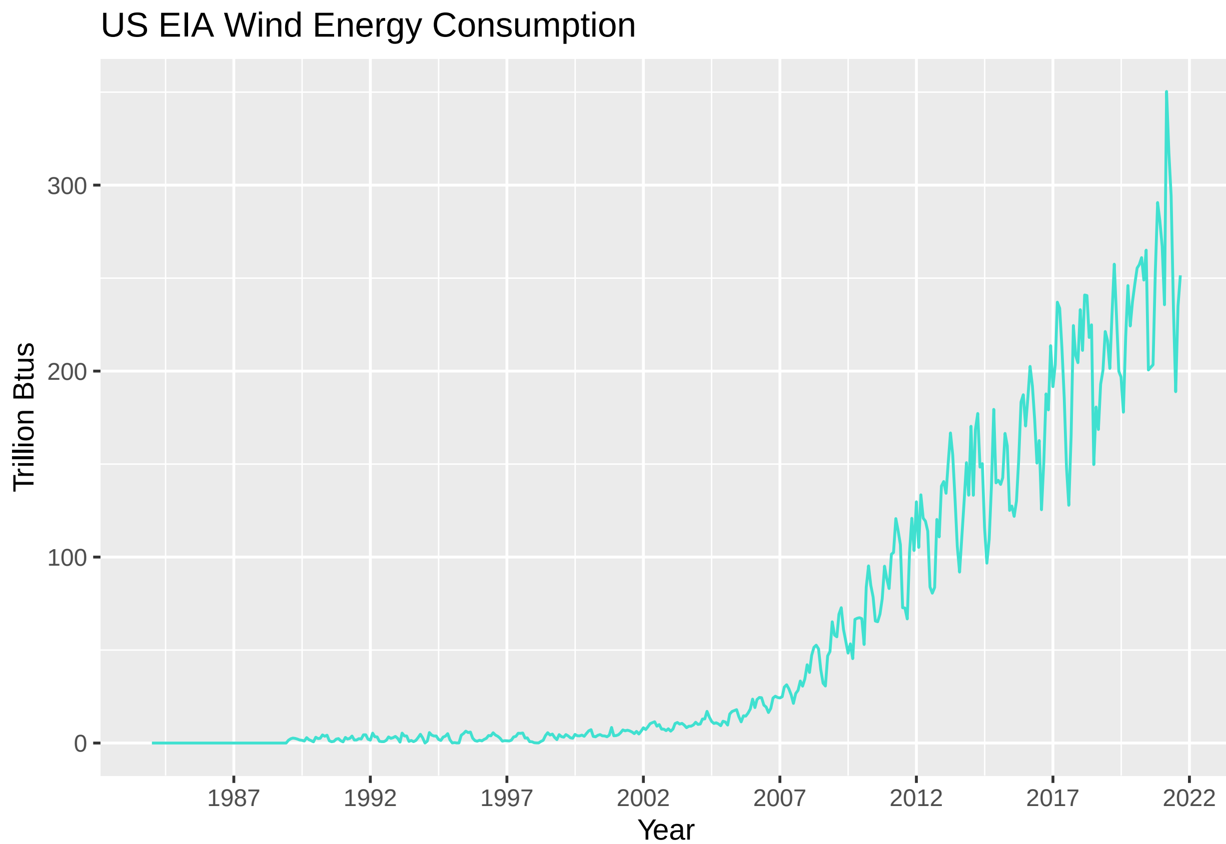
Plot the Solar and Wind energy consumption over time using ggplot. Plot each series on a separate graph. No need to add legend. Add informative names to the y axis using `ylab()`. Explore the function scale_x_date() on ggplot and see if you can change the x axis to improve your plot. Hint: use *scale_x_date(date_breaks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

```
#plot solar consumption
ggplot(energy_data3, aes(x=as.Date(Month), y=Solar_Energy_Consumption)) +
          geom_line(color="orange") +
          ggtitle("US EIA Solar Energy Consumption") +
          xlab("Year") +
          ylab(paste0("Trillion Btus")) +
          scale_x_date(date_breaks = "5 years", date_labels= "%Y")
```




```
#plot wind consumption
ggplot(energy_data3, aes(x=as.Date(Month), y=Wind_Energy_Consumption)) +
          geom_line(color="turquoise") +
          ggtitle("US EIA Wind Energy Consumption") +
          xlab("Year") +
          ylab(paste0("Trillion Btus")) +
```

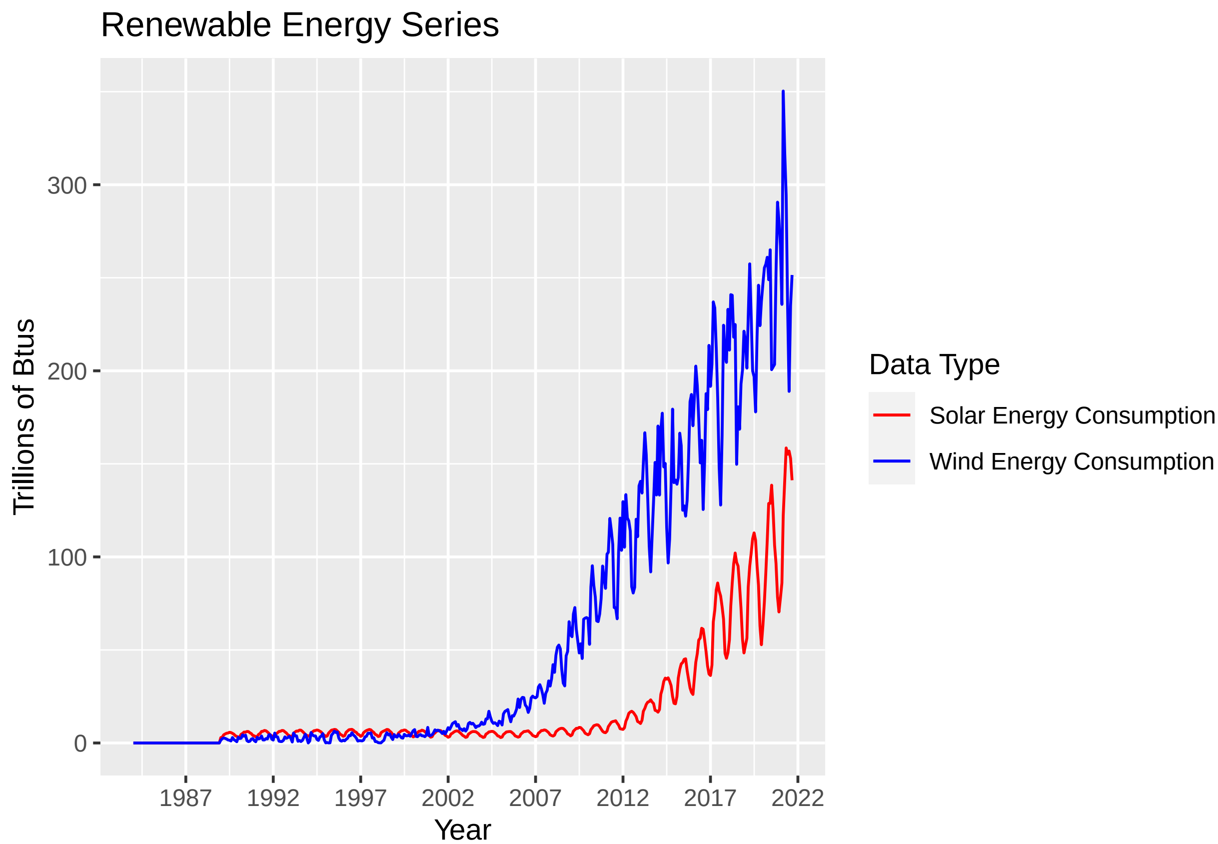
```
        scale_x_date(date_breaks = "5 years", date_labels= "%Y")
```



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(energy_data3, aes(x=as.Date(Month), y=energy_data3[,2])) +
            geom_line(aes(color="red")) +
            geom_line(aes(y=energy_data3[,3], color="blue")) +
            scale_color_identity(name="Data Type",
                                 breaks=c("red","blue"),
                                 labels=c("Solar Energy Consumption","Wind Energy Consumption"),
                                 guide = "legend") +
            labs(title="Renewable Energy Series",
                 x="Year",
                 y="Trillions of Btus",
                 ) +
           scale_x_date(date_breaks = "5 years", date_labels= "%Y")
```

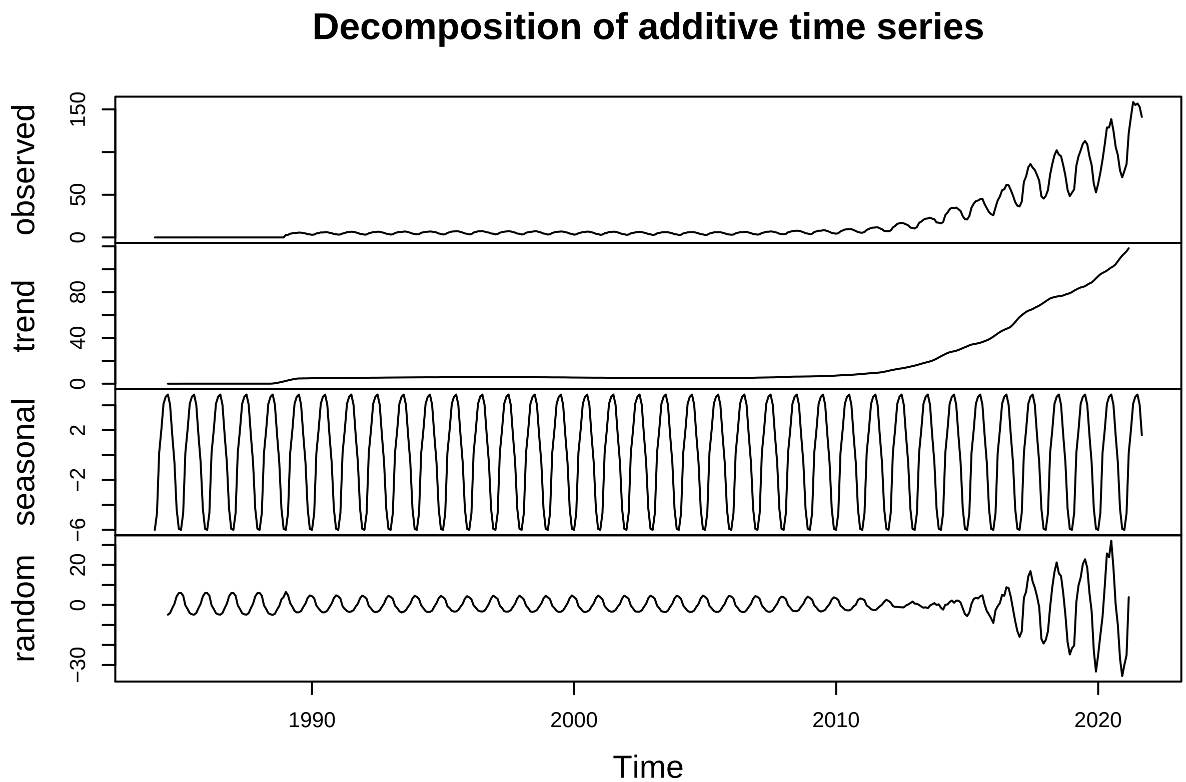


Q4

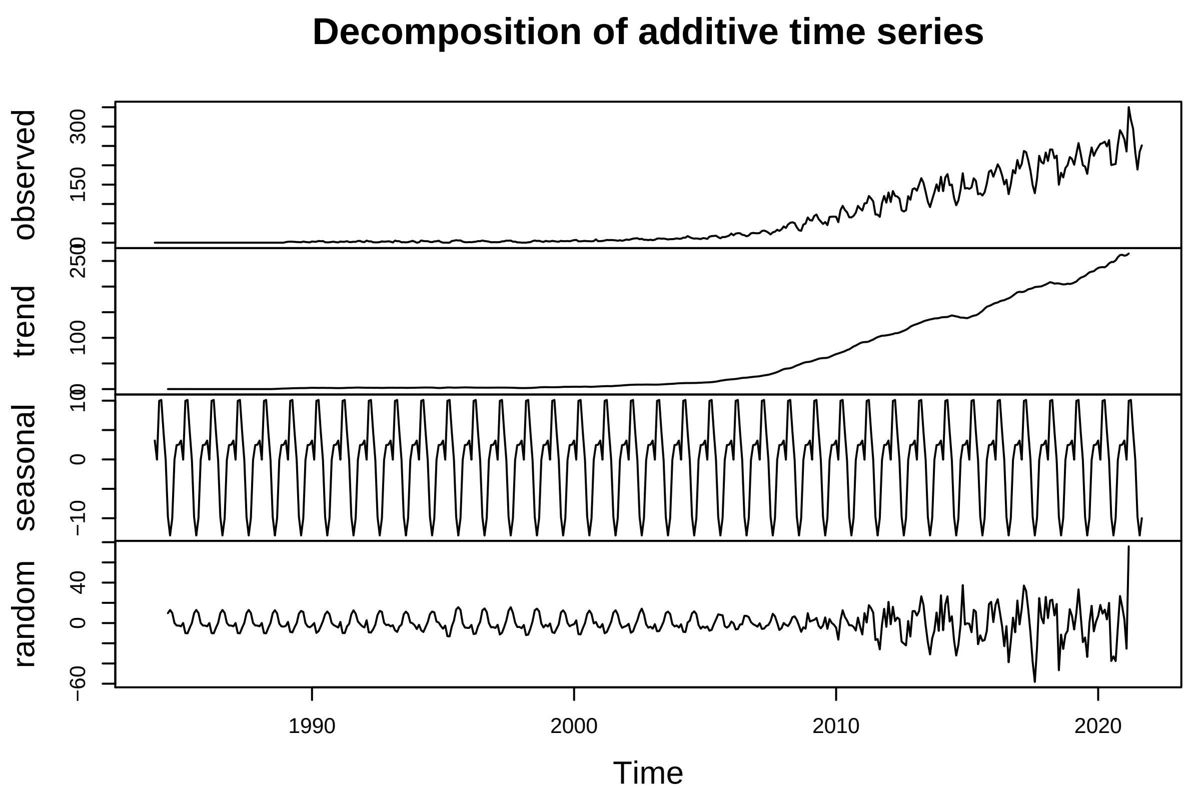
Transform wind and solar series into a time series object and apply the decompose function on them using the additive option, i.e., `decompose(ts_data, type = "additive")`. What can you say about the trend component? What about the random component? Does the random component look random? Or does it appear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

```
#transform to time series
ts_solar<-ts(energy_data3[,2],start=c(1984,1),end=c(2021,9),frequency=12)
ts_wind<-ts(energy_data3[,3],start=c(1984,1),end=c(2021,9),frequency=12)


#decompose function
decompose_solar <- decompose(ts_solar,type= "additive")
plot(decompose_solar)
```




```r
decompose_wind <- decompose(ts_wind,type= "additive")
plot(decompose_wind)
```

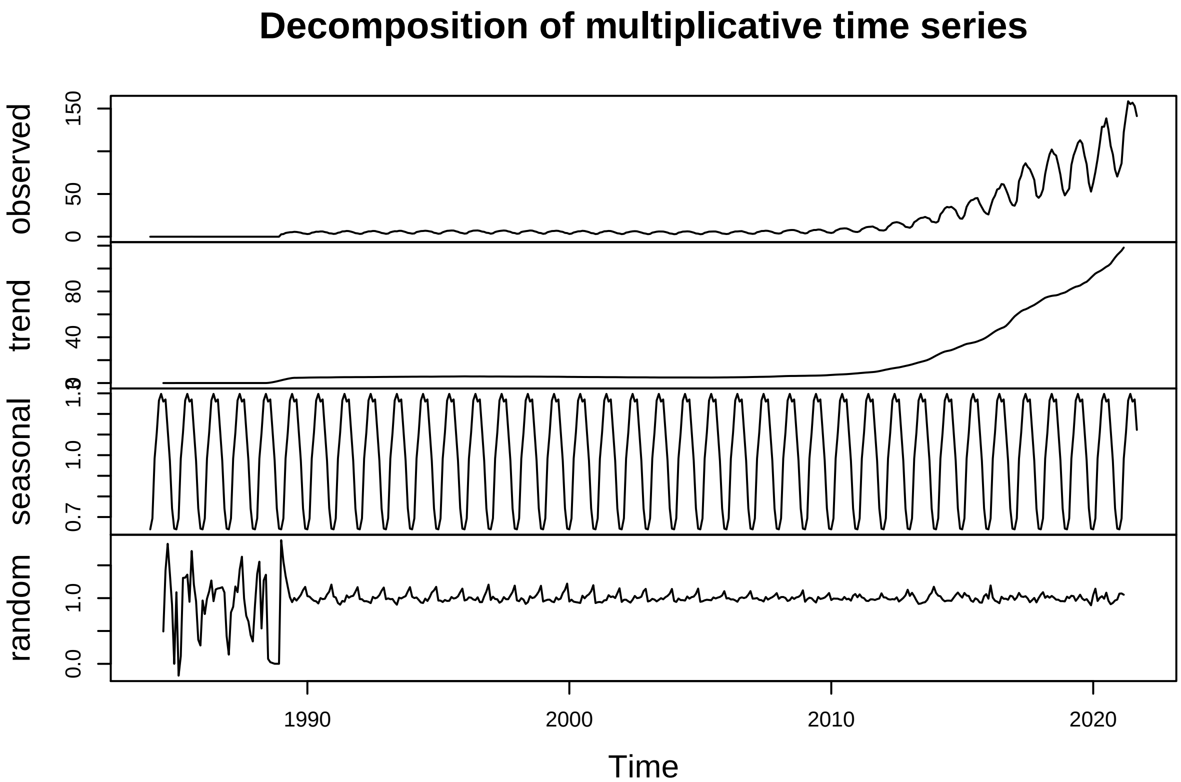


Answer: Both wind and solar are clearly upward trending. Regarding the random components, there appears to be seasonality for both solar and wind's random components because as the magnitude of the time series increases, the seasonality variation of the randoms also appears to increase as the data enters the 2010's. The random component for solar appears to have an increase in the seasonal variation as the peaks and troughs are growing larger while the seasonal variation for wind appears a little more constant. Both however though appear to experience a level shift.

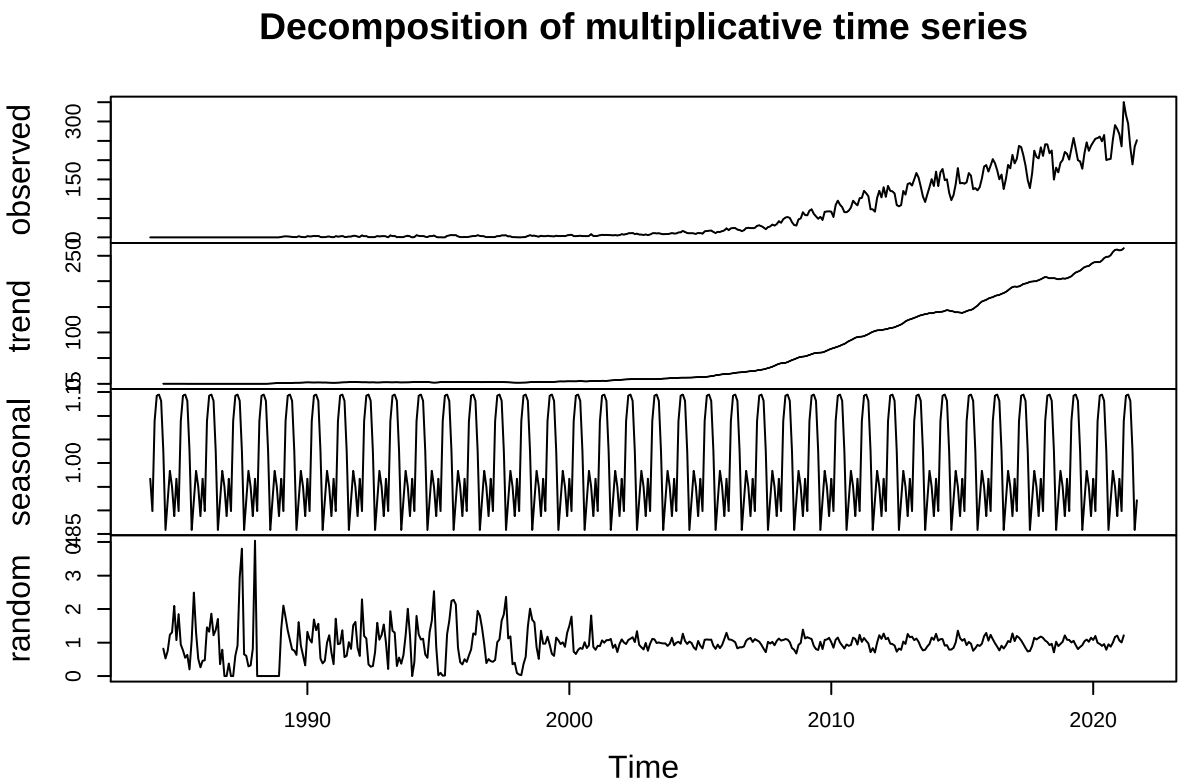
Q5

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?

```
#decompose function
decompose_solar2 <- decompose(ts_solar,type= "multiplicative")
plot(decompose_solar2)
```




```
decompose_wind2 <- decompose(ts_wind,type= "multiplicative")
plot(decompose_wind2)
```



Answer: The seasonality of the random component shifted earlier compared to later under the additive option. The random component for both solar and wind have changed in the sense that there's more seasonality variation earlier in time and then the seasonality of the random component remains constant later. Solar's random component has more variation in its seasonality up until 1990, then the seasonality remains relatively constant. Before the random component appears more constant, there is appears to be a level shift that precedes that.

Q6

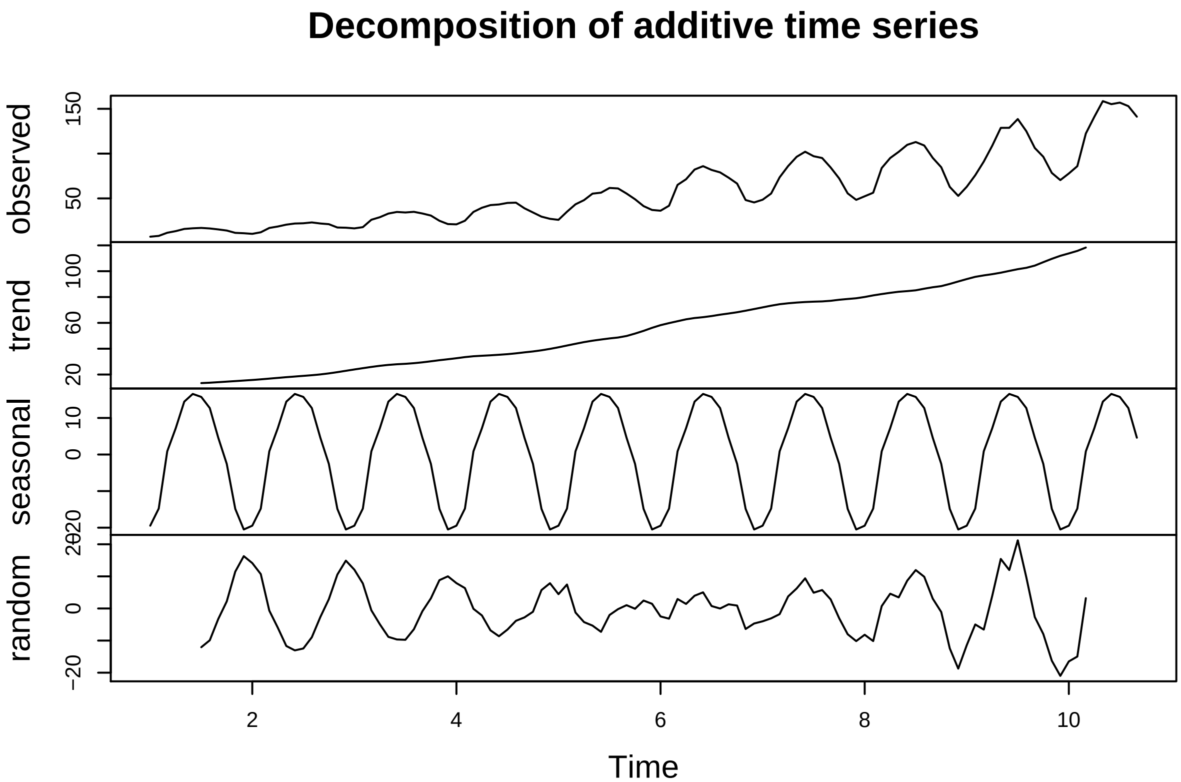
When fitting a model to this data, do you think you need all the historical data? Think about the data from 90s and early 20s. Are there any information from those years we might need to forecast the next six months of Solar and/or Wind consumption. Explain your response.

> Answer: I think data from the 90s for both solar/wind would be unnecessary because wind technology wasn't really developed in depth until the 2000s and solar was being slowly developed during the 1990s. Incorporating this data would affect the trend of the time series and cause an underestimation of forecasts of future renewable energy consumption. Data from the early 1990s would be considered a level shift outlier as although the seasonal behavior hasn't changed, the level of the time series during the given period has changed. Solar/wind consumption will continue to rise and incorporating data from the 90s where it's barely being developed would alter the trend.

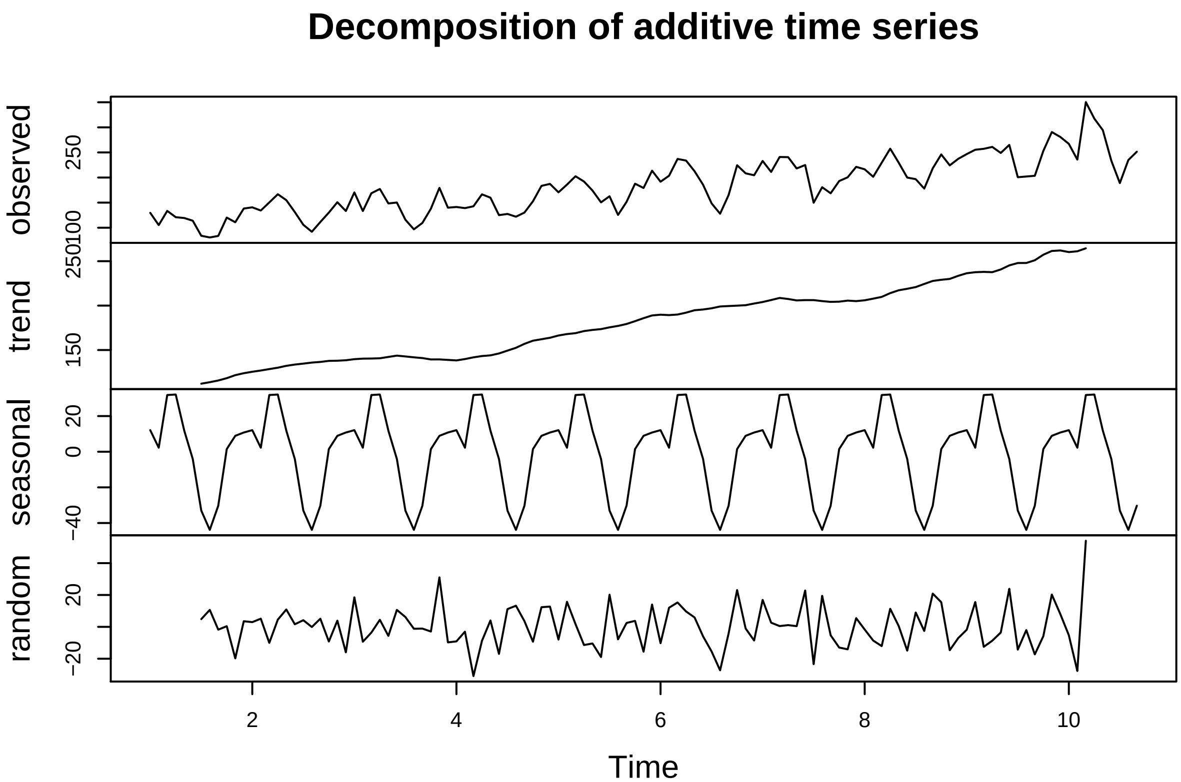
Q7

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.

```
energy_data4 <- filter(energy_data3, year(Month)>=2012)
new_solar.ts<-ts(energy_data4$Solar_Energy_Consumption, frequency = 12)
new_wind.ts<-ts(energy_data4$Wind_Energy_Consumption, frequency = 12)
decompose_new_solar <- decompose(new_solar.ts, type = "additive")
decompose_new_wind <- decompose(new_wind.ts, type = "additive")
plot(decompose_new_solar)
```




```
plot(decompose_new_wind)
```



Answer: The random component appears to have outliers have better controlled for compared to the previous plots as we don't see as drastic of level shifts in the data. The random component data also appears to have a more constant mean compared to the other plots. Filtering the outliers out has helped to make the seasonal part of the data look less distorted. This will help produce a more accurate forecast of the data. However, there is of course still some seasonality to the random components for both solar and wind as there's still peaks and troughs and these peaks/troughs especially for solar almost correlate with the seasonal component ex. a trough for solar's seasonal component is roughly a peak for the random component.