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

Assignment 5 - Due date 02/28/22

Rob Kravec

```
library(readxl)
library(forecast)
library(tseries)
library(Kendall)
library(lubridate)
library(tidyverse)
library(patchwork)
```

Decomposing Time Series

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

```
# Importing data set - using readxl package
path <- paste0("../Data/Table_10.1_Renewable_Energy_Production_and",
               "_Consumption_by_Source.xlsx")
energy_data <- read_excel(path = path,
                          col_names = FALSE, skip = 12, sheet = "Monthly Data",
                          na = "Not Available")

# Now let's extract the column names from row 11 only
read_col_names <- read_excel(path = path,
                             col_names = FALSE, skip = 10 , n_max =1,
                             sheet = "Monthly Data")

# Add column names to energy_data df
colnames(energy_data) <- read_col_names
head(energy_data)
```

```
## # A tibble: 6 x 14
##   Month               `Wood Energy Production` `Biofuels Produ~ `Total Biomass ~
##   <dttm>                                 <dbl>            <dbl>            <dbl>
## 1 1973-01-01 00:00:00                     130.               NA             130.
## 2 1973-02-01 00:00:00                     117.               NA             117.
## 3 1973-03-01 00:00:00                     130.               NA             130.
## 4 1973-04-01 00:00:00                     125.               NA             126.
## 5 1973-05-01 00:00:00                     130.               NA             130.
## 6 1973-06-01 00:00:00                     125.               NA             126.
## # ... with 10 more variables: Total Renewable Energy Production <dbl>,
## #   Hydroelectric Power Consumption <dbl>, Geothermal Energy Consumption <dbl>,
## #   Solar Energy Consumption <dbl>, Wind Energy Consumption <dbl>,
## #   Wood Energy Consumption <dbl>, Waste Energy Consumption <dbl>,
```

```
## #   Biofuels Consumption <dbl>, Total Biomass Energy Consumption <dbl>,
## #   Total Renewable Energy Consumption <dbl>
```

```
# Count rows and columns in energy_data df
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!

```
# Make desired data frame
df1 <- energy_data %>%
  rename(Solar = "Solar Energy Consumption",
         Wind = "Wind Energy Consumption") %>%
  select(Month, Solar, Wind) %>%
  drop_na(any_of(c("Solar", "Wind")))

# Display first 6 rows
head(df1)
```

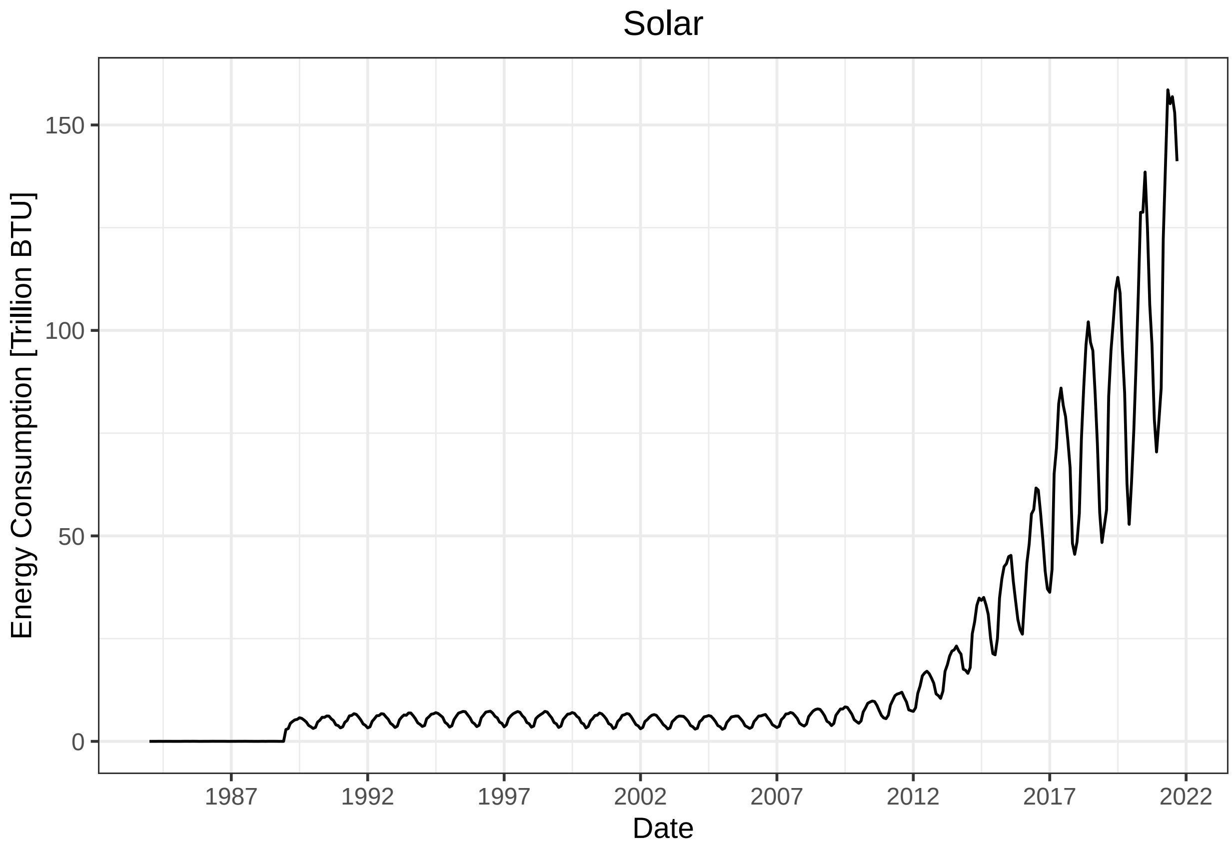
```
## # A tibble: 6 x 3
##   Month                Solar  Wind
##   <dttm>               <dbl> <dbl>
## 1 1984-01-01 00:00:00 -0.001 0
## 2 1984-02-01 00:00:00  0.001 0.002
## 3 1984-03-01 00:00:00  0.002 0.002
## 4 1984-04-01 00:00:00  0.003 0.006
## 5 1984-05-01 00:00:00  0.007 0.008
## 6 1984-06-01 00:00:00  0.01  0.006
```

It's strange that a couple of `Solar Energy Consumption` values are negative (-0.001 Trillion BTU). These values are quite infrequent (and probably just measurement errors?), so I won't worry too much about them.

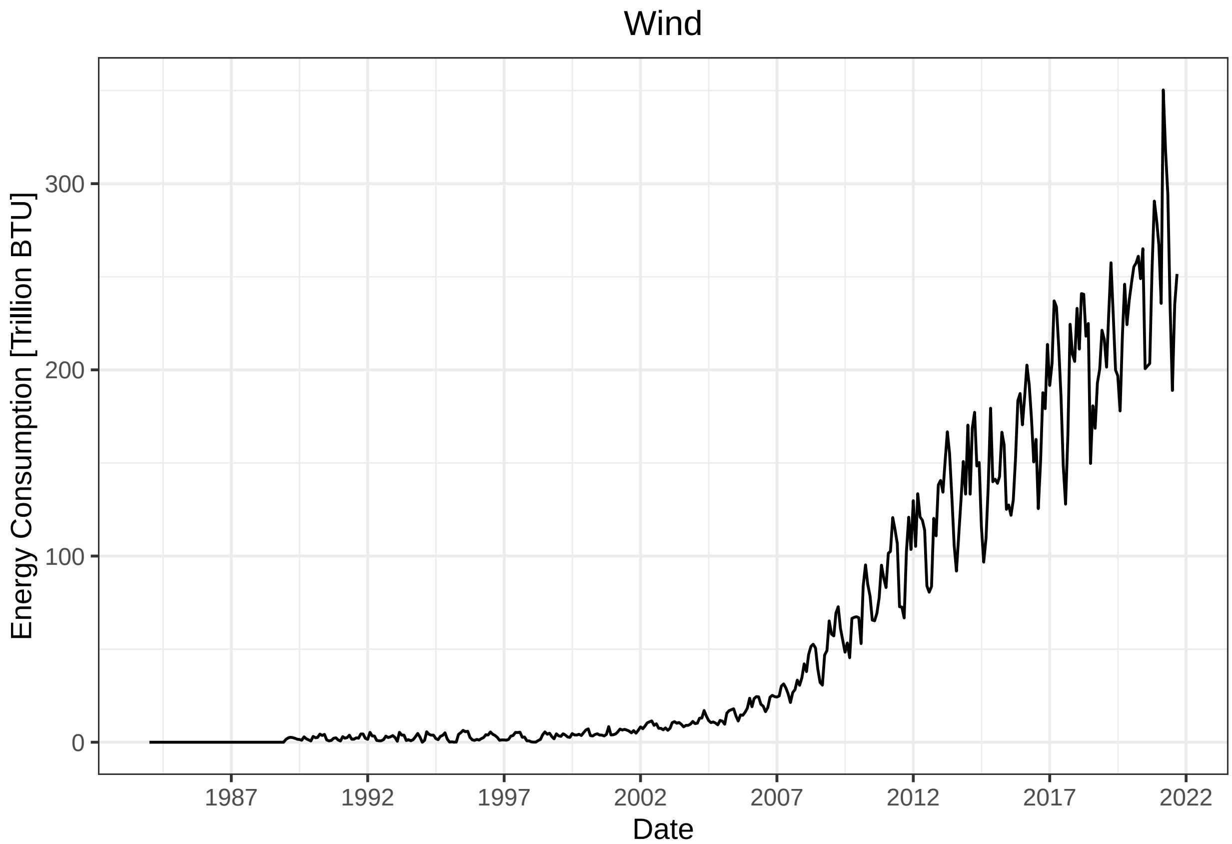
Q2

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
ggplot(data = df1, mapping = aes(x = Month, y = Solar)) +
  geom_line() +
  theme_bw() +
  labs(x = "Date", y = "Energy Consumption [Trillion BTU]", title = "Solar") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_x_datetime(date_breaks = "5 years", date_labels = "%Y")
```



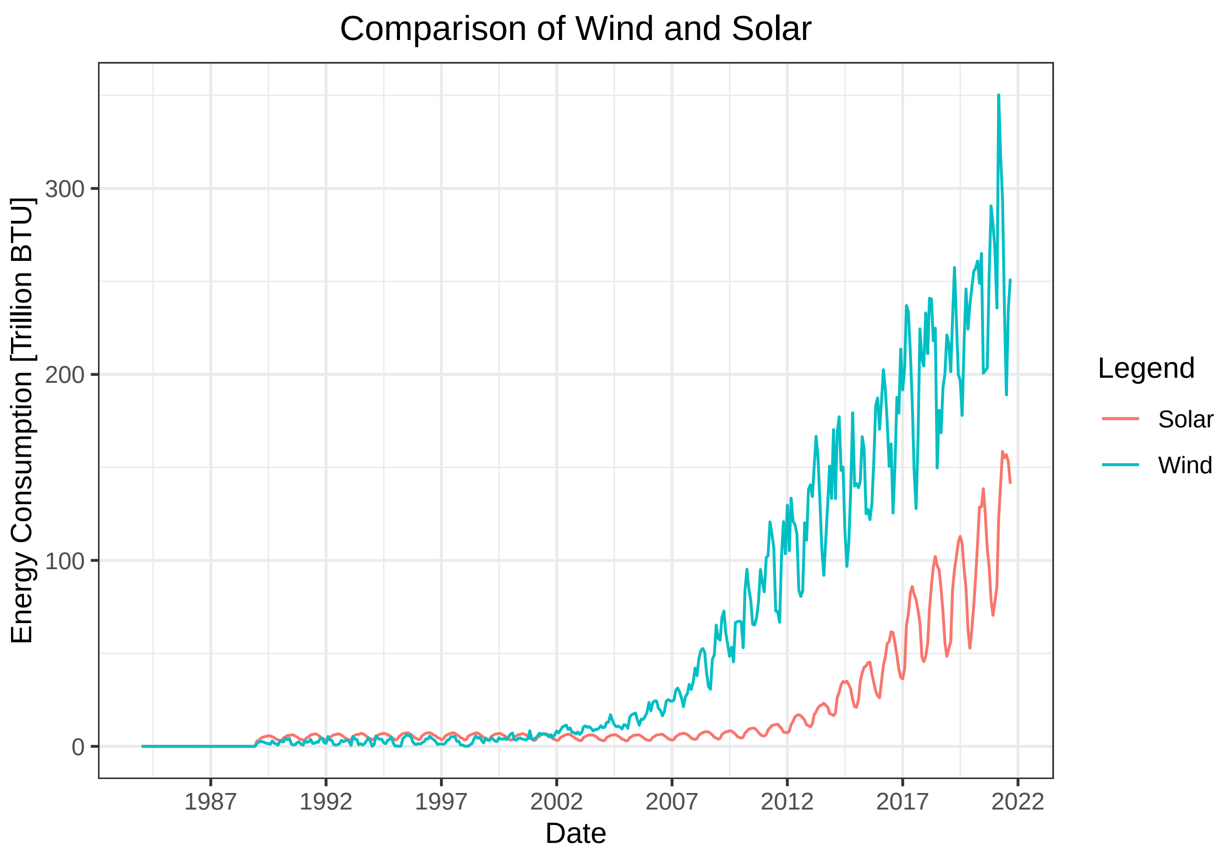

```r
# Plot Wind
ggplot(data = df1, mapping = aes(x = Month, y = Wind)) +
  geom_line() +
  theme_bw() +
  labs(x = "Date", y = "Energy Consumption [Trillion BTU]", title = "Wind") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_x_datetime(date_breaks = "5 years", date_labels = "%Y")
```



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(data = df1, mapping = aes(x = Month)) +
  geom_line(mapping = aes(y = Solar, color = "Solar")) +
  geom_line(mapping = aes(y = Wind, color = "Wind")) +
  theme_bw() +
  labs(x = "Date", y = "Energy Consumption [Trillion BTU]",
       title = "Comparison of Wind and Solar",
       color = "Legend") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_x_datetime(date_breaks = "5 years", date_labels = "%Y")
```



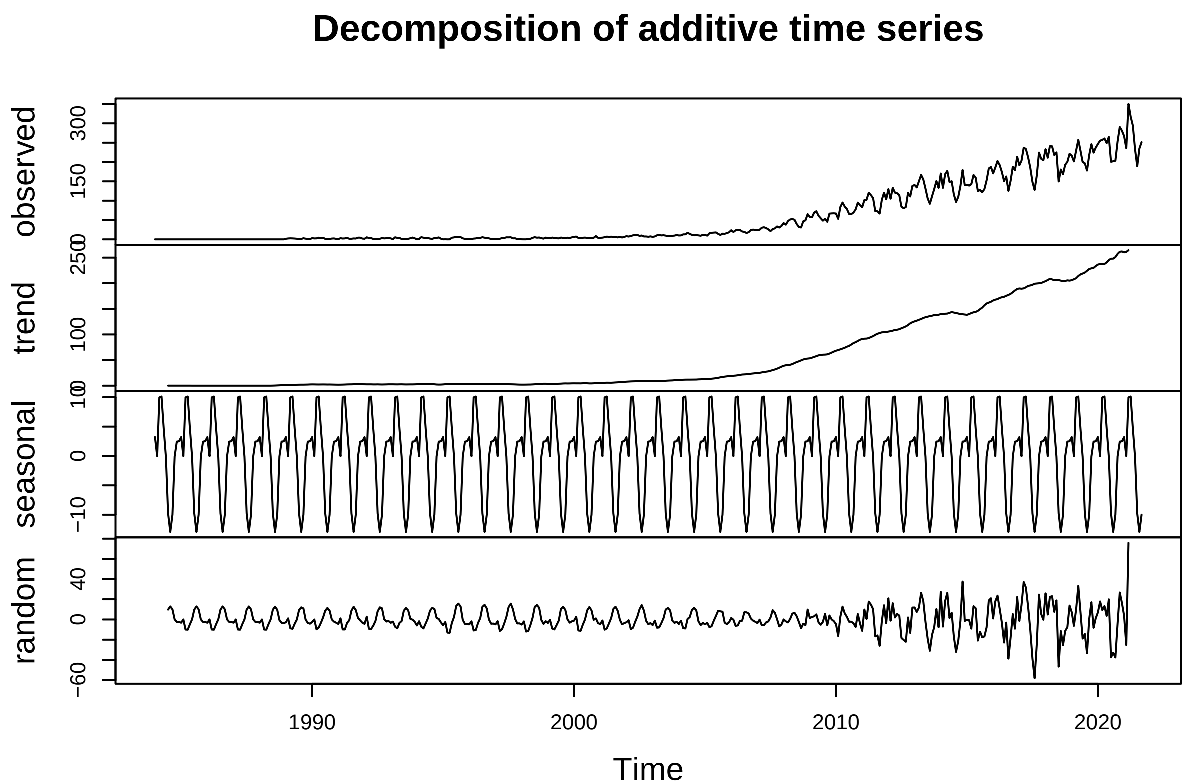
Q4

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?

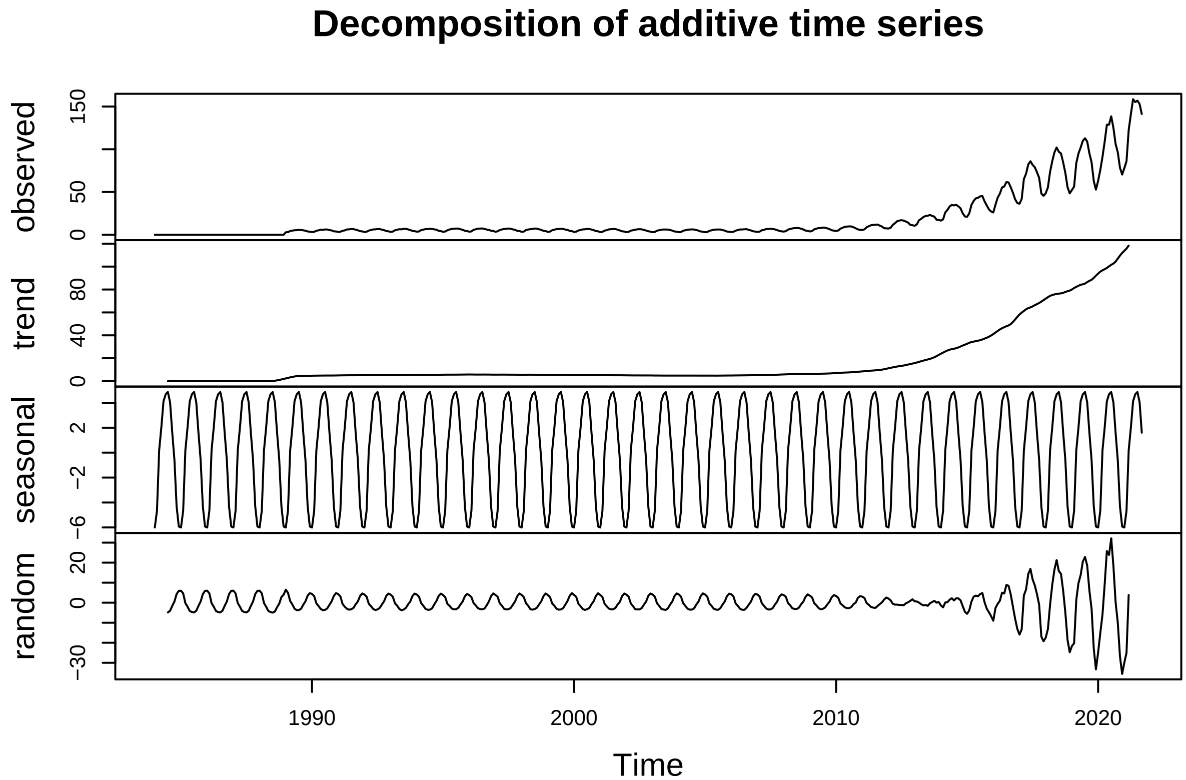
```
# Create time series objects
wind_ts <- ts(data = df1$Wind, frequency = 12, start = c(1984, 1))
solar_ts <- ts(data = df1$Solar, frequency = 12, start = c(1984, 1))

# Decompose time series
wind_decompose <- decompose(wind_ts)
solar_decompose <- decompose(solar_ts)
```

The `Wind` trend component shows a strong upward trajectory starting around 2007, while the `Solar` trend component shows a strong upward trajectory starting at around 2012. Both random components definitely show signs of seasonality with a wave-like appearance. Additionally, the magnitude of the random component for both time series has increased substantially in recent years (post-2010 for `Wind` and post-2015 for `Solar`).

```
# Plot results
plot(wind_decompose)
```




```
plot(solar_decompose)
```



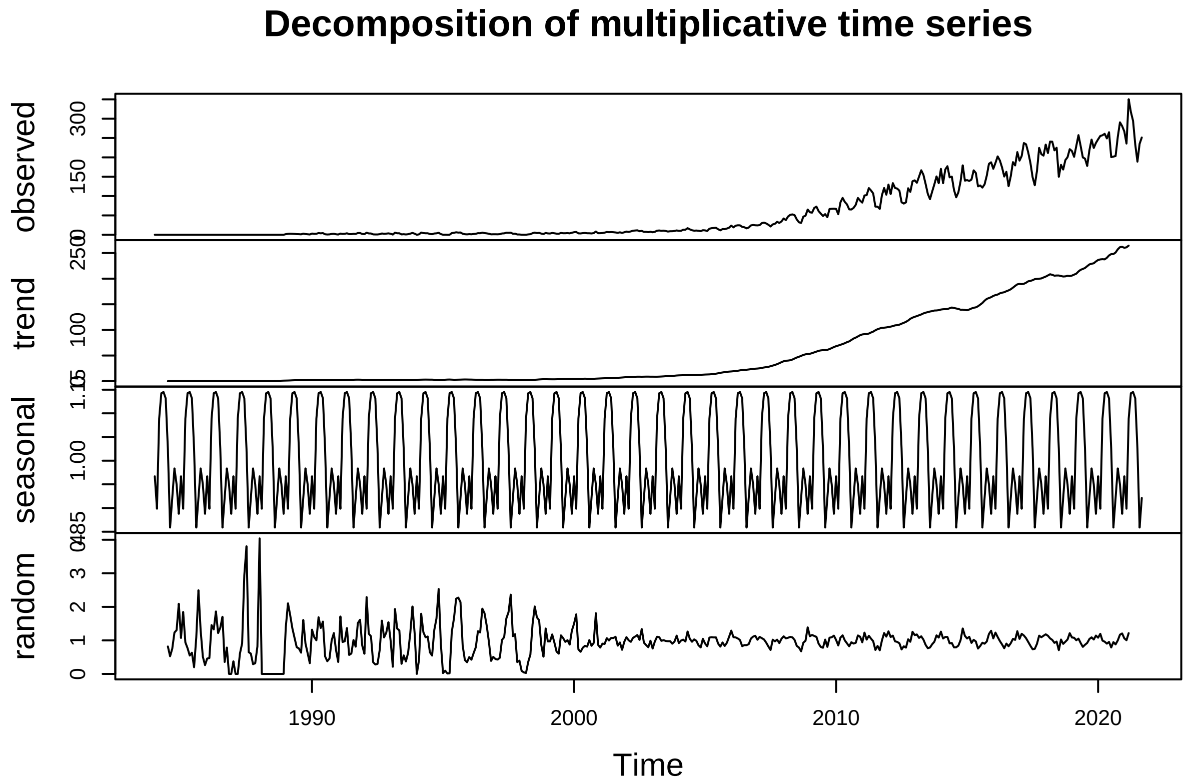
Q5

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?

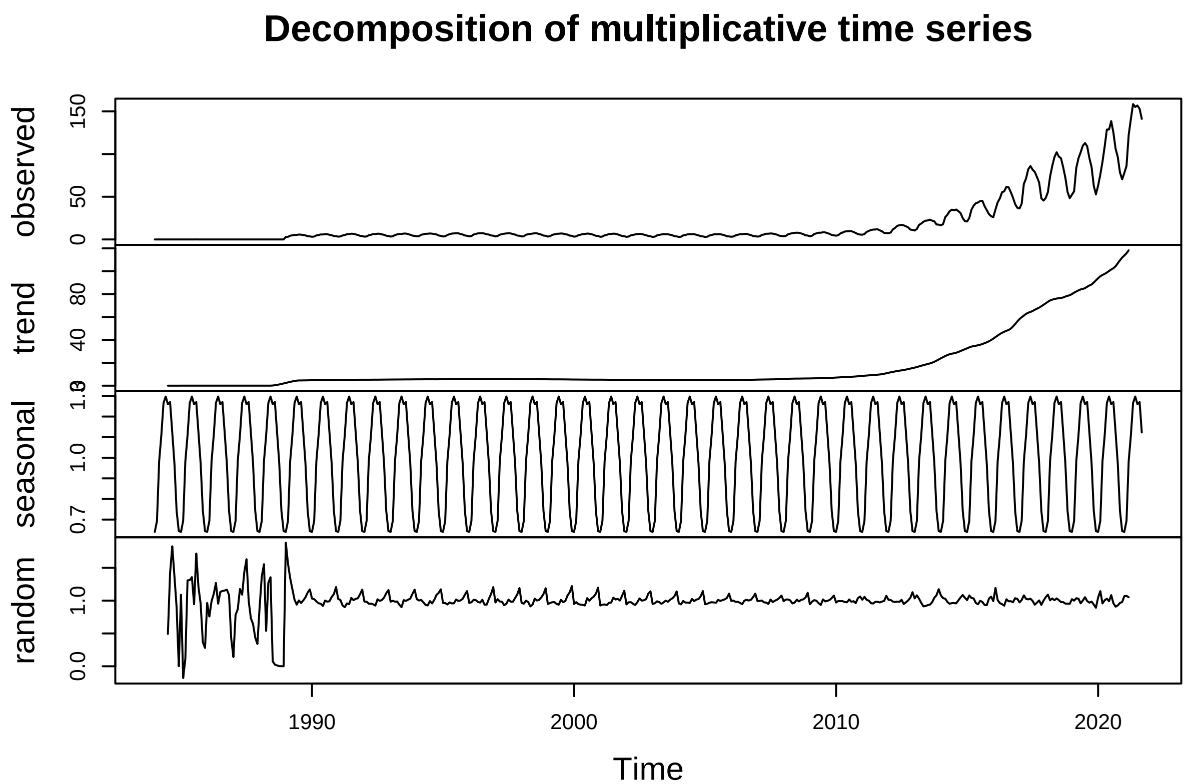
```
# Decompose time series
wind_decompose2 <- decompose(wind_ts, type = "multiplicative")
solar_decompose2 <- decompose(solar_ts, type = "multiplicative")
```

Both random components still show some signs of seasonality. For example, the random component for `Wind` looks wave-like from about 2005-2020. The random component for `Solar` looks wave-like from about 1990-2005. That said, the multiplicative time series decomposition is a definite improvement over the additive decomposition. The seasonal nature of the random component is much less pronounced and occurs over a smaller percentage of the time frame considered.

```
# Plot results
plot(wind_decompose2)
```




```
plot(solar_decompose2)
```



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.

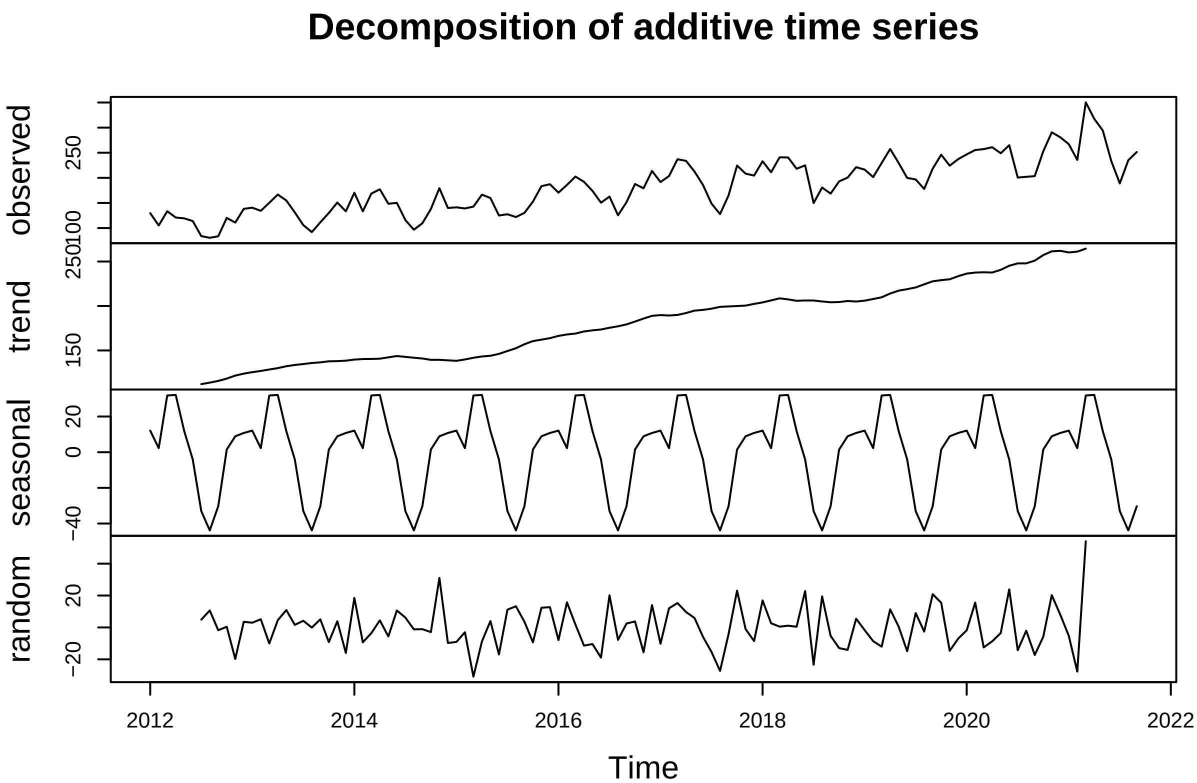
> Answer: No, I don't think we need all of this historical data to forecast the next six months of Solar and/or Wind consumption. We only need the data that will provide meaningful information about what to expect over the next six month period. As an extreme example of why we don't need to include all of the historical data, consider the years 1984-2002. Energy consumption is close to zero for both Solar and Wind, lending very little information about present consumption of these energy media. I imagine there is a significant trade-off between (1) wanting to use very recent data for projections and (2) having enough data to make the projections meaningful. The next exercise has us filter the dataset to start in January 2012 – perhaps that would make sense for modeling Wind, while the less mature growth trajectory of solar could warrant a later start date (e.g., January 2017).

Q7

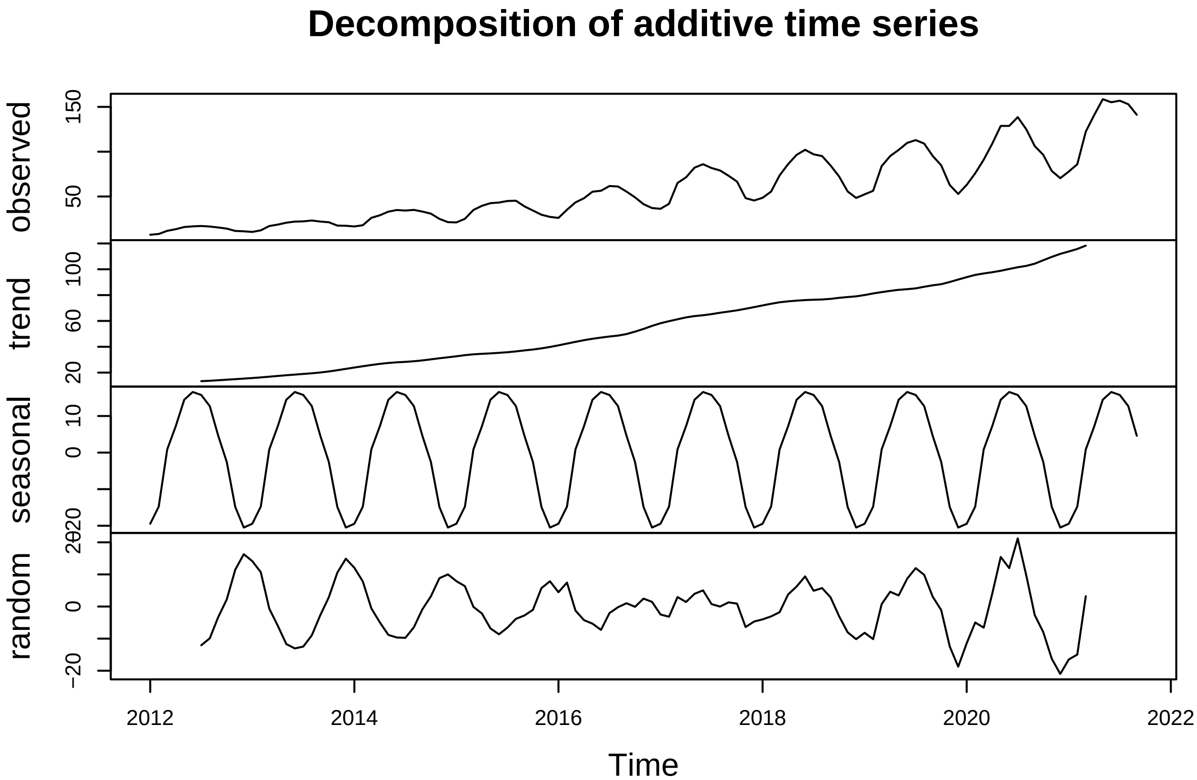
Create a new time series object where historical data starts on January 2012. Hint: use `filter()` function so that you don't need to point to row numbers, .i.e, `filter(xxxx, year(Date) >= 2012 )`. Apply the decompose function `type=additive` to this new time series. Comment the results. Does the random 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.

```
# Perform desired filtering
df7 <- df1 %>%
```

```
  filter(year(Month) >= 2012)

# Create new time series objects
wind_ts3 <- ts(data = df7$Wind, frequency = 12, start = c(2012, 1))
solar_ts3 <- ts(data = df7$Solar, frequency = 12, start = c(2012, 1))

# Decompose time series
wind_decompose3 <- decompose(wind_ts3)
solar_decompose3 <- decompose(solar_ts3)

# Plot results
plot(wind_decompose3)
```




```
plot(solar_decompose3)
```



Answer: The random component for the Wind time series looks pretty random, which is great progress. On the other hand, the random component for the Solar time series still appears to have some seasonality. This result is not entirely surprising because, starting in 2012, the Wind time series has a relatively steady (i.e., linear) upward trajectory, while the Solar time series has positive concavity. Since the decompose function models the seasonal component as constant over time, it is able to model the Seasonal component pretty well for the Wind time series and quite poorly for the Solar time series.