ENV 797 - Time Series Analysis for Energy and Environment Applications | Spring 2025

Assignment 5 - Due date 02/18/25

Aye Nyein Thu

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_Sp25.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: "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)
library(tseries)
library(ggplot2)
library(Kendall)
library(lubridate)
library(tidyverse)  #load this package so yon clean the data frame using pipes
library(openxlsx)
library(dplyr)
```

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 December 2023 Monthly Energy Review.

```
#Importing data set - using xlsx package
energy_data <- read.xlsx(
  xlsxFile = "./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
  colNames = FALSE, startRow = 13, sheet = 1) #startRow is equivalent to skip on read.table
```

```r
#Now let's extract the column names from row 11 only
read_col_names <- read.xlsx(
  xlsxFile = "./Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx",
  colNames = FALSE, rows = 11, sheet = 1)

colnames(energy_data) <- read_col_names
head(energy_data)

energy_data$Month <- as.Date(energy_data$Month, origin = "1900-01-01")

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!

```r
# Check Not Available in the columns of interest
sum(energy_data$"Solar Energy Consumption" =="Not Available")
```

```
## [1] 132
```

```r
sum(energy_data$"Wind Energy Consumption" =="Not Available")
```

```
## [1] 120
```

```r
# Select the columns of interest and drop NA
energy_cleaned <- energy_data %>%
  select(
    Date = Month,
    Solar = "Solar Energy Consumption",
    Wind = "Wind Energy Consumption") %>%
  filter(
    Solar != "Not Available" &
      Wind != "Not Available") %>%
  mutate(
    Solar = as.numeric(Solar),
    Wind = as.numeric(Wind)
  ) %>%
  drop_na()

# Recheck Not Available in the columns of interest
sum(energy_cleaned$Solar =="Not Available")
```

```
## [1] 0
```

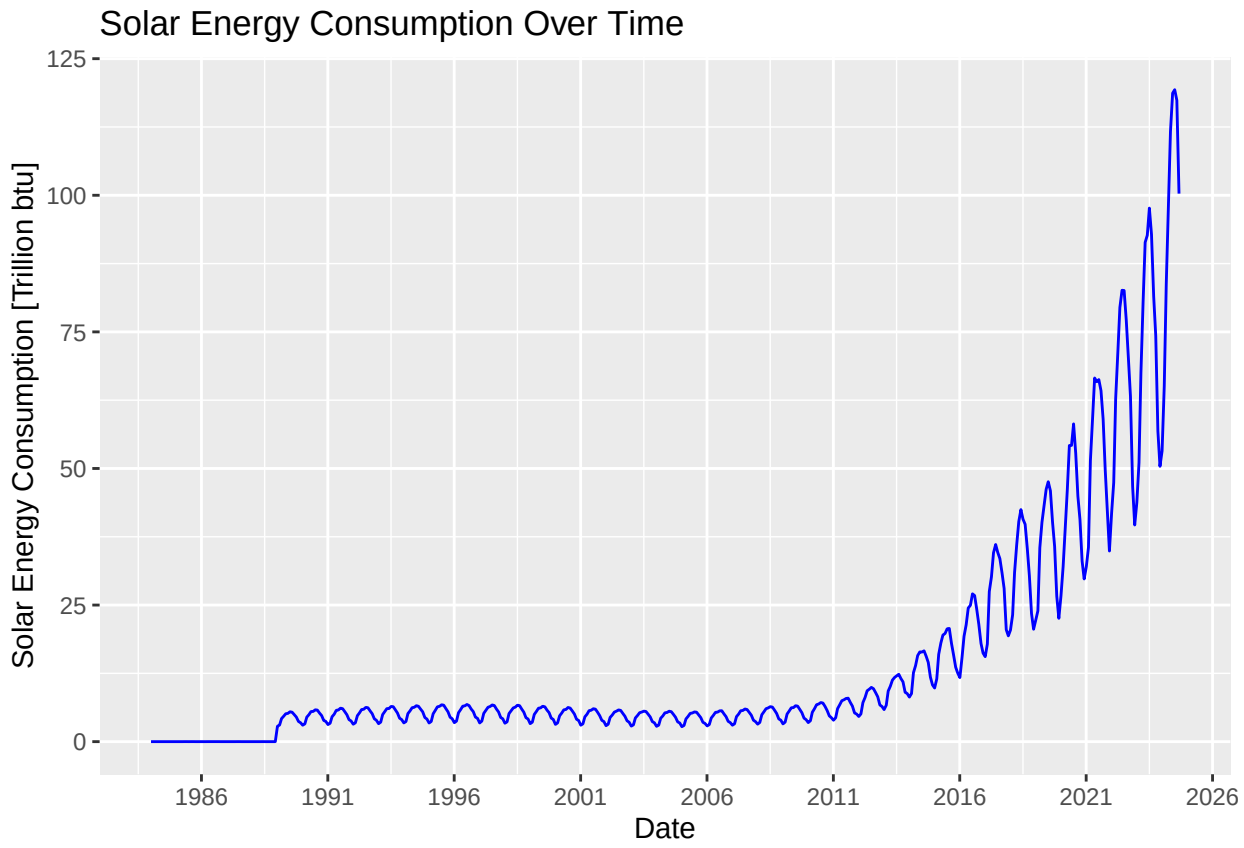
```r
sum(energy_cleaned$Wind =="Not Available")
```

```
## [1] 0
```

Q2

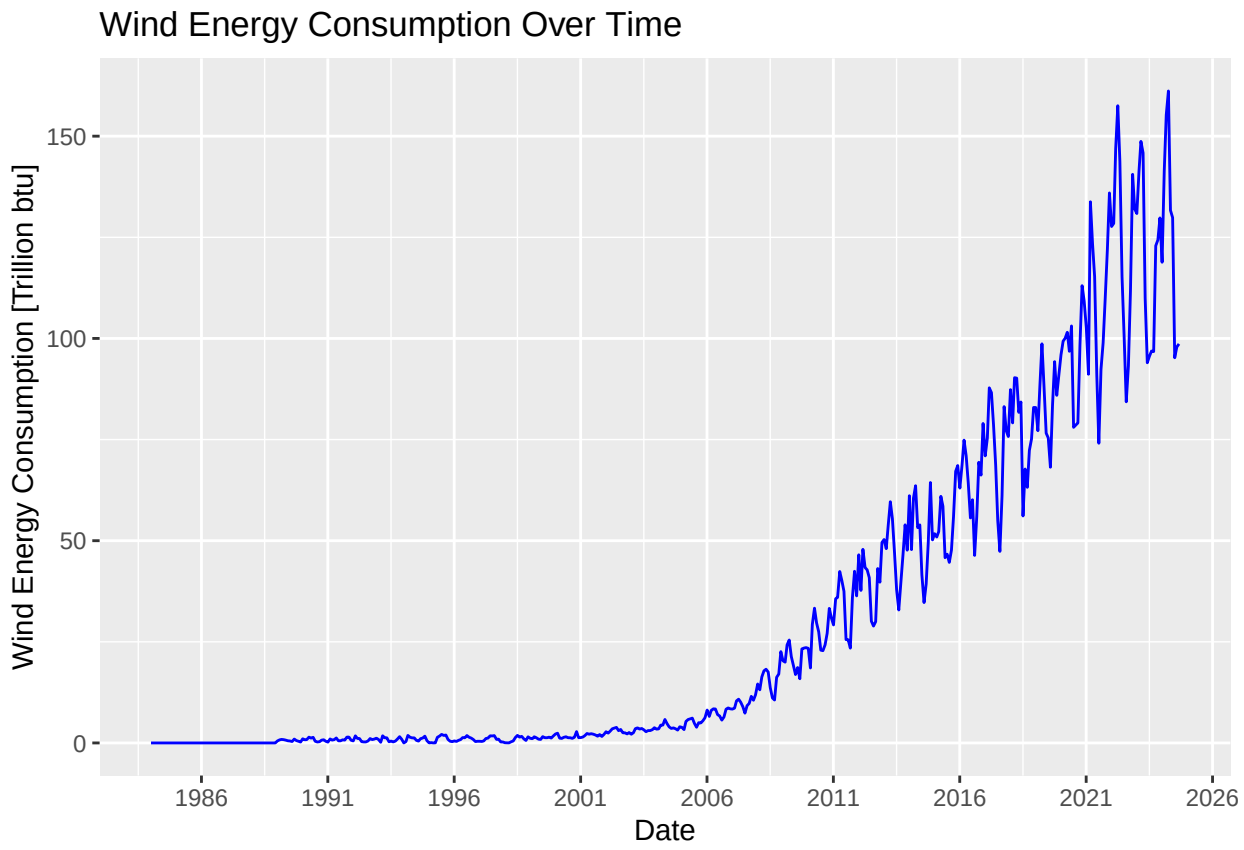
Plot the Solar and Wind energy consumption over time using ggplot. Plot each series on a separate graph. No need to add legend. Add informative names to the y axis using `ylab()`. Explore the function `scale_x_date()` on ggplot and see if you can change the x axis to improve your plot. Hint: use `scale_x_date(date_breaks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

```r
# Plot Solar Energy Consumption
ggplot(energy_cleaned, aes(x = Date, y = Solar)) +
  geom_line(color = "blue") +
  ylab("Solar Energy Consumption [Trillion btu]") +
  scale_x_date(date_breaks = "5 years",
               date_labels = "%Y") +
  ggtitle("Solar Energy Consumption Over Time")
```




```r
# Plot Wind Energy Consumption
ggplot(energy_cleaned, aes(x = Date, y = Wind)) +
  geom_line(color = "blue") +
  ylab("Wind Energy Consumption [Trillion btu]") +
  scale_x_date(date_breaks = "5 years",
```

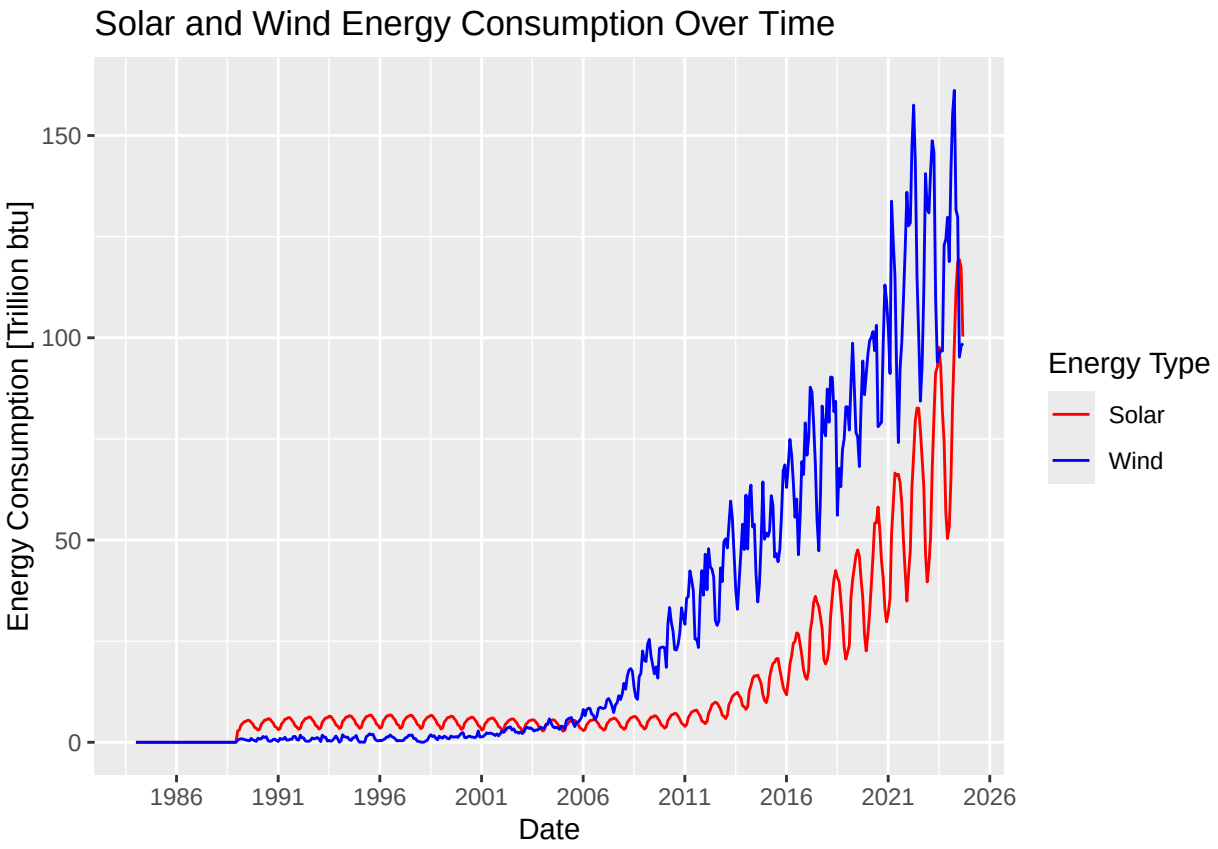
```
             date_labels = "%Y") +
  ggtitle("Wind Energy Consumption Over Time")
```



Q3

Now plot both series in the same graph, also using ggplot(). Use function `scale_color_manual()` 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()` to set x axis breaks every 5 years.

```
# Plot two series in one graph
ggplot(energy_cleaned, aes(x = Date)) +
  geom_line(aes(y = Solar, color = "Solar")) +
  geom_line(aes(y = Wind, color = "Wind")) +
  scale_color_manual(values = c("Solar" = "red",
                                "Wind" = "blue")) +
  labs(color = "Energy Type") +
  ylab("Energy Consumption [Trillion btu]") +
  scale_x_date(date_breaks = "5 years",
               date_labels = "%Y") +
  ggtitle("Solar and Wind Energy Consumption Over Time")
```



Decomposing the time series

The stats package has a function called decompose(). This function only take time series object. As the name says the decompose function will decompose your time series into three components: trend, seasonal and random. This is similar to what we did in the previous script, but in a more automated way. The random component is the time series without seasonal and trend component.

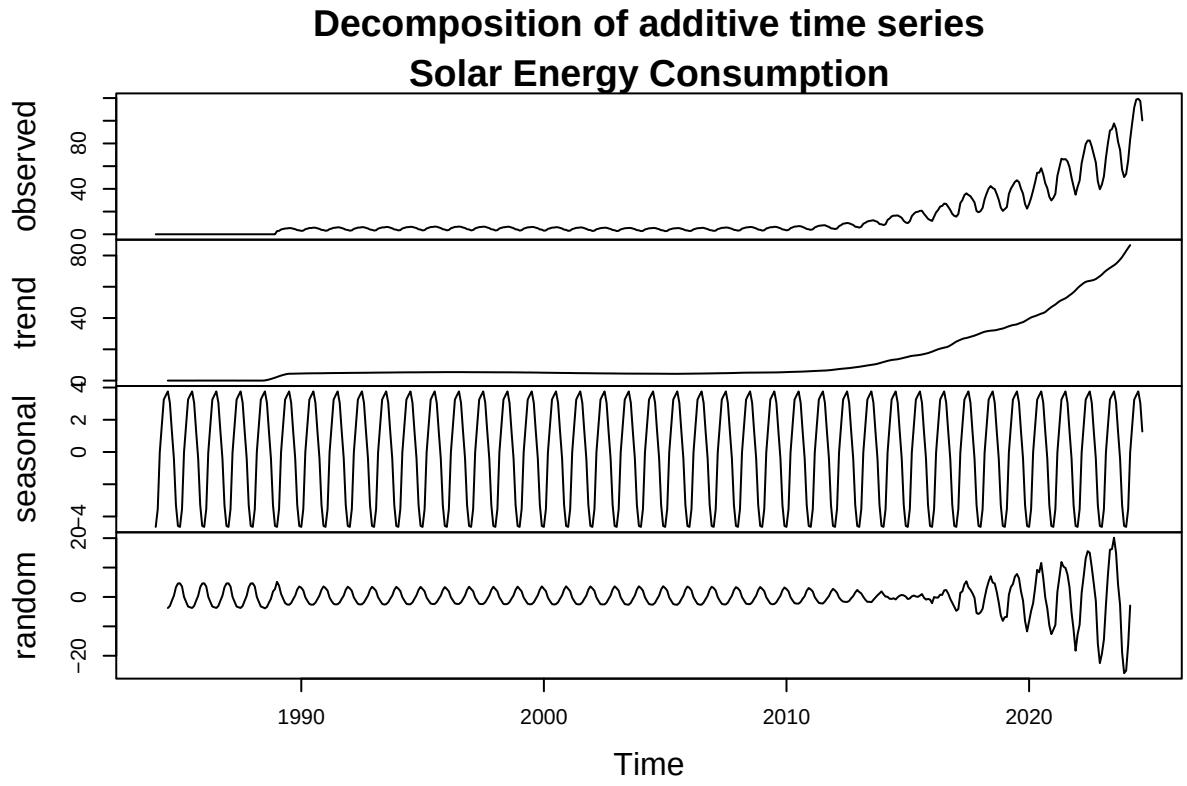
Additional info on `decompose()`.

1) You have two options: alternative and multiplicative. Multiplicative models exhibit a change in frequency over time.
2) The trend is not a straight line because it uses a moving average method to detect trend.
3) The seasonal component of the time series is found by subtracting the trend component from the original data then grouping the results by month and averaging them.
4) The random component, also referred to as the noise component, is composed of all the leftover signal which is not explained by the combination of the trend and seasonal component.

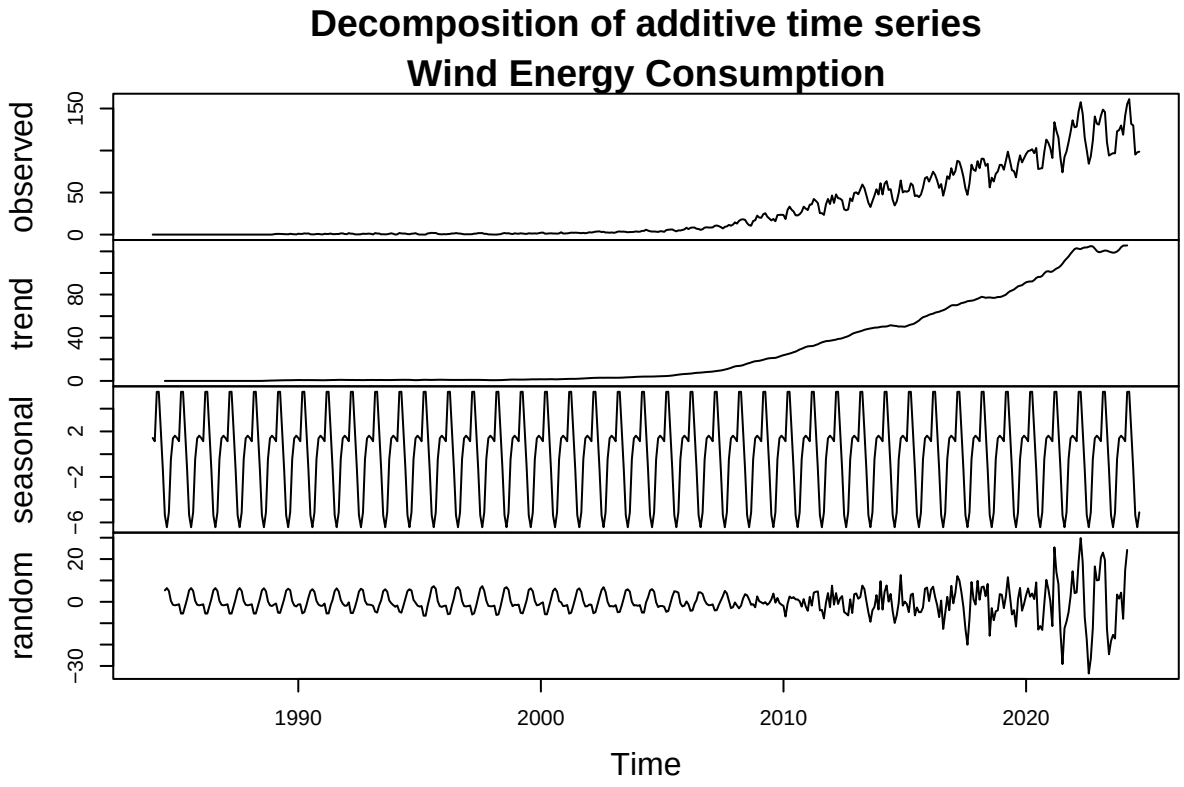
Q4

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?

```r
# Specify starting points
start_year <- as.numeric(format(min(energy_cleaned$Date), "%Y"))
start_month <- as.numeric(format(min(energy_cleaned$Date), "%m"))

# Transform data frame to time series objects
ts_energy <- ts(energy_cleaned[,2:3],
                start = c(start_year, start_month), frequency = 12)

# Decompose solar energy series with additive option
decompose_add_solar <- decompose(ts_energy[,1],"additive")

# Plot the solar energy series
plot(decompose_add_solar)
title(main = "Solar Energy Consumption", line = 1, outer = FALSE)
```




```r
# Decompose wind energy series with additive option
decompose_add_wind <- decompose(ts_energy[,2],"additive")

# Plot the wind energy series
plot(decompose_add_wind)
title(main = "Wind Energy Consumption", line = 1, outer = FALSE)
```



Answer: For both solar and wind energy consumption, the series have a mix of stable and trend episodes. For the solar energy series, the data points are stable for the period between 1984 to around 2012; after that, it exhibits a strong increasing trend. Similarly, the wind energy series also have a constant data period from 1984 to around 2006 and the following period represents a strong upward trend. The "trend" component in both plots are able to reflect these elements of constant and increasing trends and it could be assume that moving average estimate is effective.

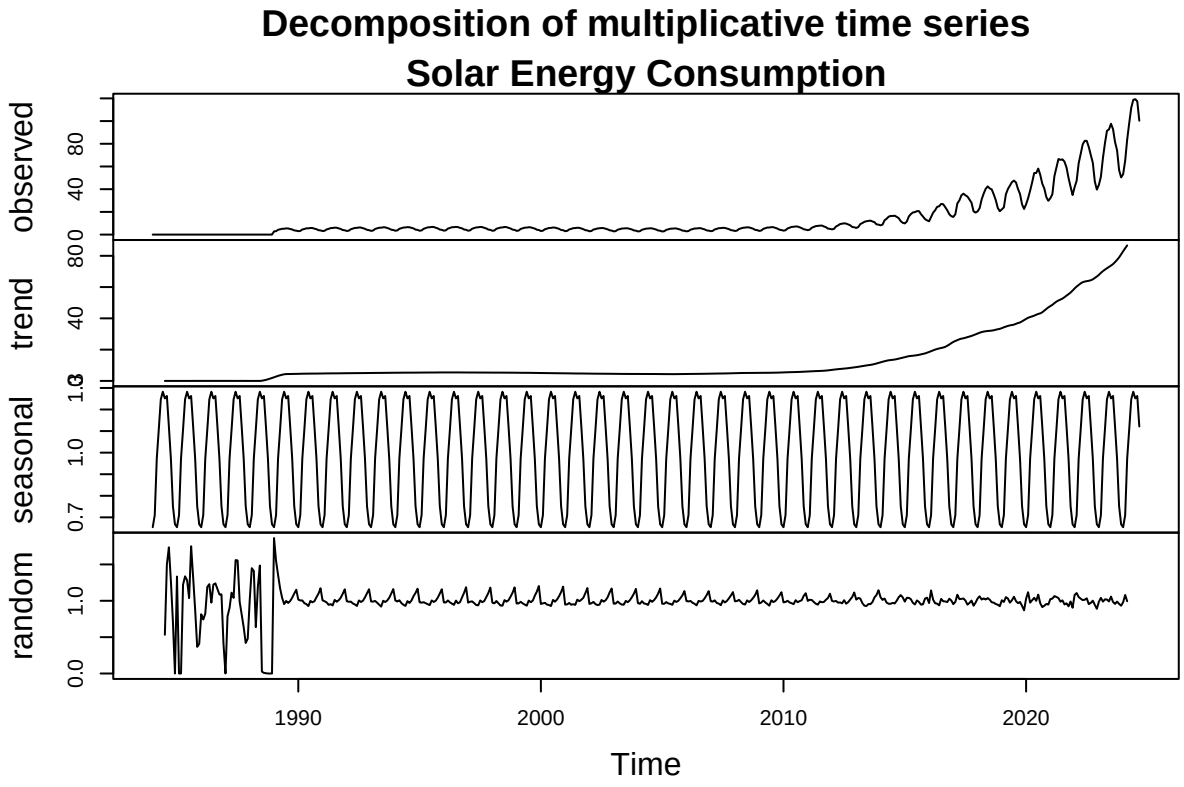
Further, the "seasonal" component in both plots of the series indicate a regular ups and downs, therefore, it is likely that both solar and wind energy have seasonality behaviours.

When it comes to "random" component, the seasonality-alike features seem to be still identifiable in solar energy due to some regular ups and downs in the series. For the wind energy series, it also exhibits a slight seasonality features in the initial historial data and more in randoms after that period.

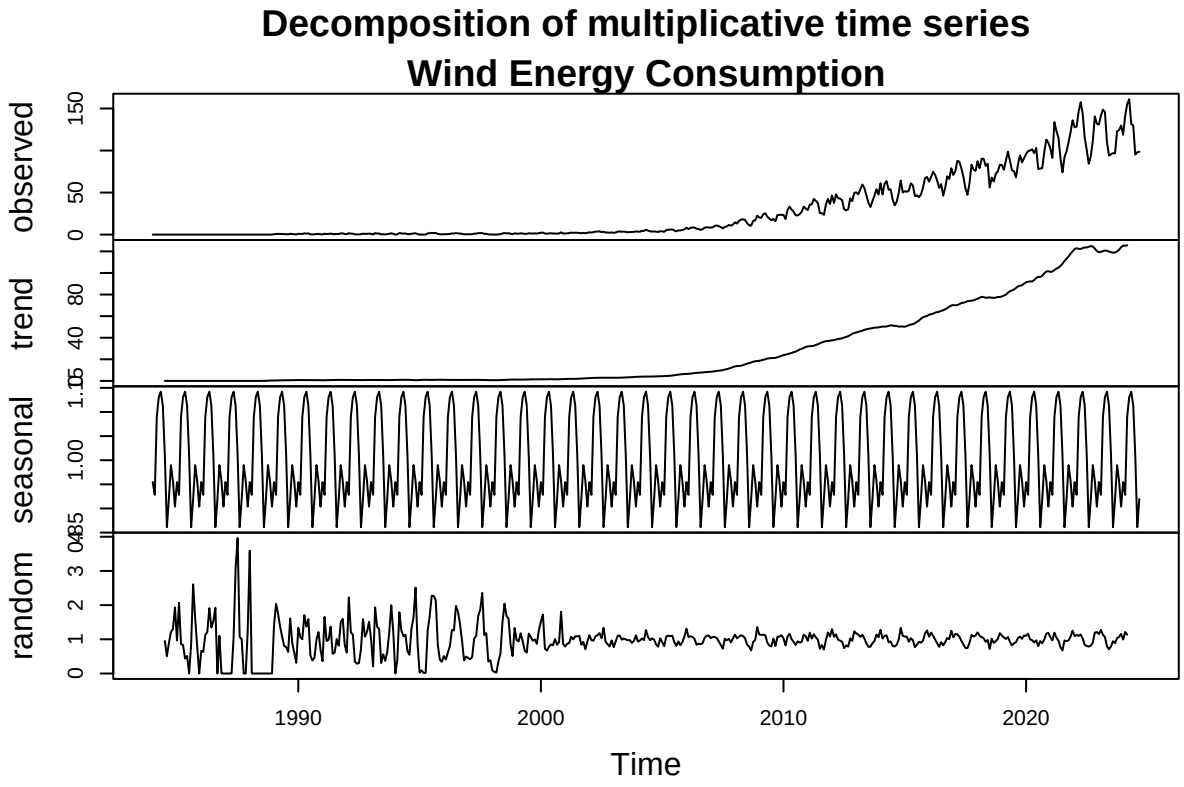
Q5

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 solar energy series with multiplicative option
decompose_multi_solar <- decompose(ts_energy[,1],"multiplicative")
plot(decompose_multi_solar)
title(main = "Solar Energy Consumption", line = 1, outer = FALSE)
```




```r
# Decompose wind energy series with multiplicative option
decompose_multi_wind <- decompose(ts_energy[,2],"multiplicative")
plot(decompose_multi_wind)
title(main = "Wind Energy Consumption", line = 1, outer = FALSE)
```



Answer: When the multiplicative method is used, the 1980s period now become randoms in nature for the solar series and the 90s period exhibit randoms in wind energy. In the multiplicative model, the random component represents the proportional deviations from the trend and seasonal components. In both series, the period of 20s to 2024 flatten out. In both Q4 and Q5, the historical data representing 1980s, 1990s and early 2000s are more or less constant and different from the increasing trend behaviors in the late 2000s until 2024 period. This constant characteristics seems to affect the random component in both data series.

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.

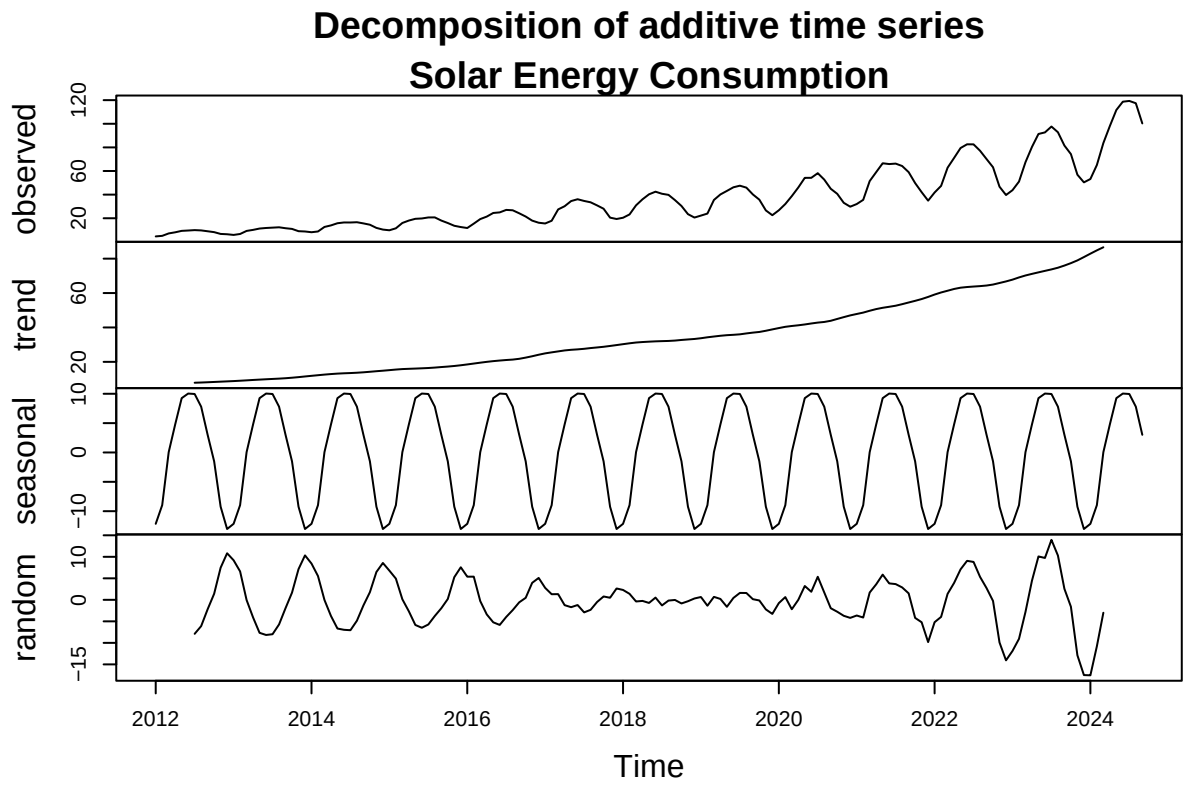
Answer: If we are to estimate the solar and wind consumption for short-term period (in this case for 6 months), incorporating the stable/ constant historial data period from 90s to early 20s into the data series might not be very useful. After this period, both series exhibit level-shift behaviours in which the change in data are not likely to go back to the original state as in 90s and early 20s. Apart from that, the estimation is only for a shorter time frame. Therefore, removing the historial data might be more useful and accurate for estimation.

Q7

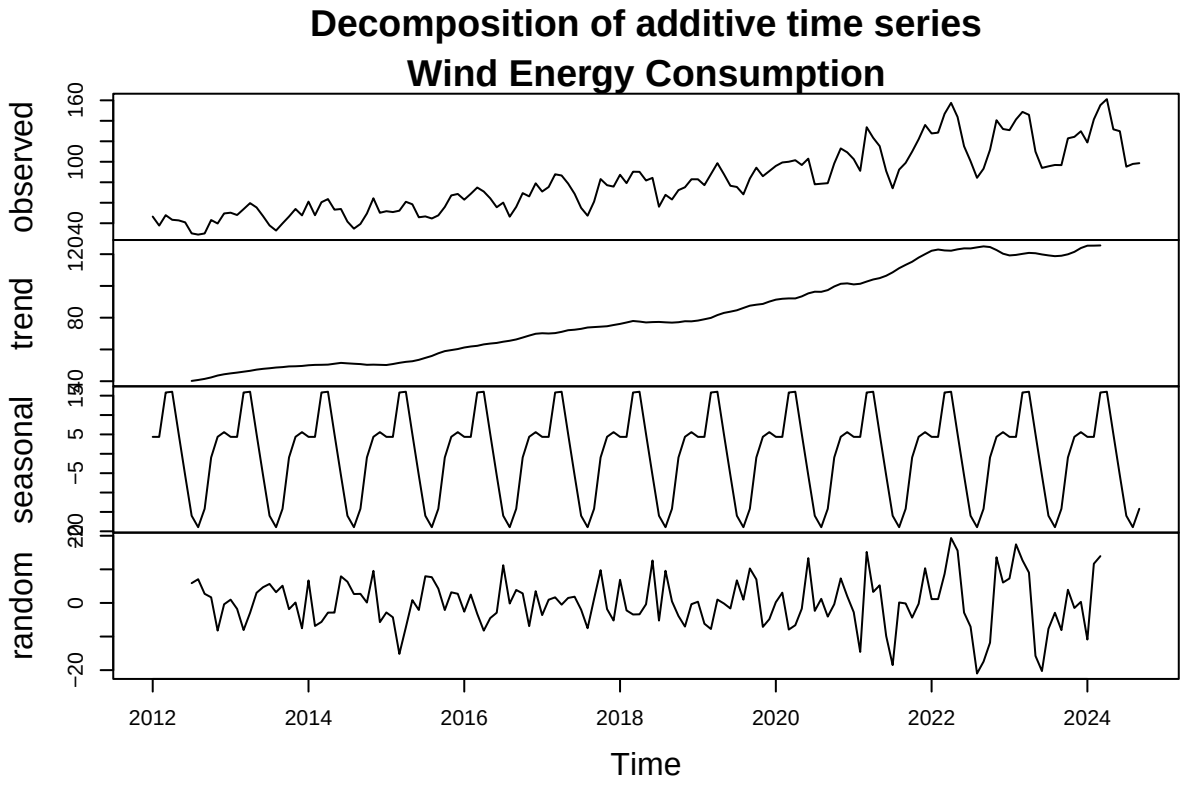
Create a new time series object where historical data starts on January 2012. Hint: use `filter()` function so that you don't need to point to row numbers, .i.e, `filter(xxxx, year(Date) >= 2012 )`. Apply the

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 seasonal components that depends on the level of the series.

```
# Filter years equal to and beyond 2012
energy_2012 <- energy_cleaned %>%
  filter(year(Date) >= 2012)

# Transform to time series object
ts_energy_2012 <- ts(energy_2012[,2:3],
                start = c(2012, 1), frequency = 12)

# Decompose solar energy series with additive option
decompose_solar_2012 <- decompose(ts_energy_2012[,1],"additive")
plot(decompose_solar_2012)
title(main = "Solar Energy Consumption", line = 1, outer = FALSE)
```




```
# Decompose wind energy series with additive option
decompose_wind_2012 <- decompose(ts_energy_2012[,2],"additive")
plot(decompose_wind_2012)
title(main = "Wind Energy Consumption", line = 1, outer = FALSE)
```



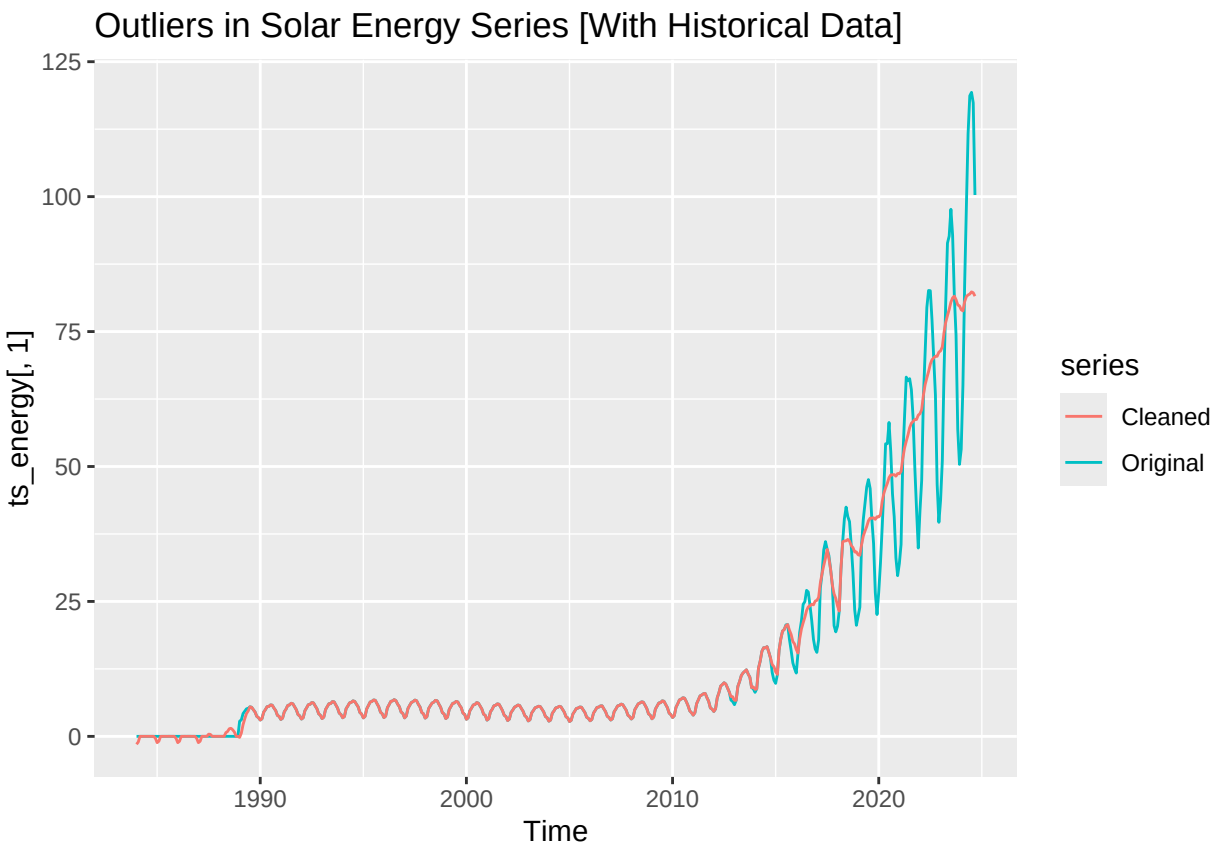
Answer: When the historical data is removed and time series objects are created from 2012, the "random" components in the wind energy plot seems to be in random nature despite some spikes. However, for the solar energy, there is still a presence of seasonality-like variations in the random component despite a slight level-off between 2018 and 2020. The solar energy might have the multiplicative feature in which seasonal components increases depending on the level of the series.

Identify and Remove outliers

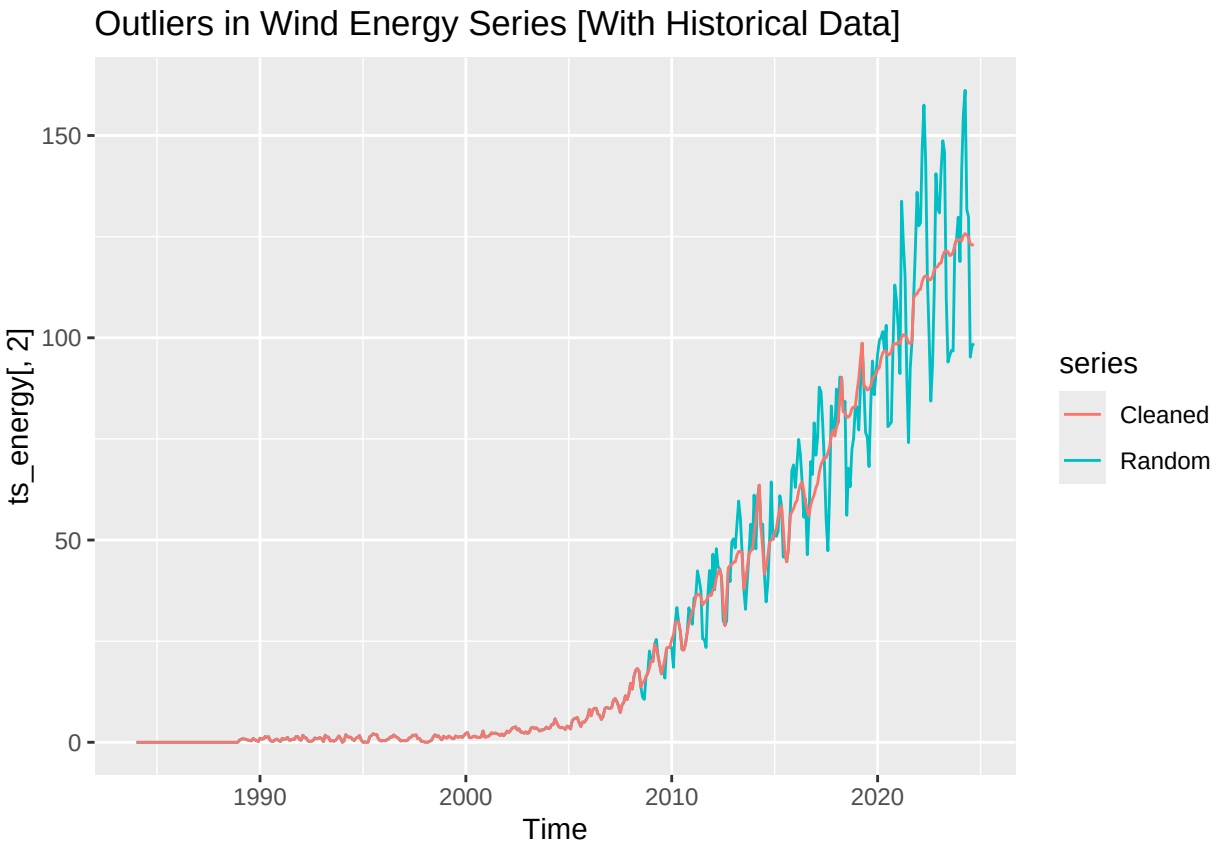
Q8

Apply the `tsclean()` to both time series object you created on Q4. Did the function removed any outliers from the series? Hint: Use `autoplot()` to check if there is difference between cleaned series and original series.

```
# Apply tsclean function for solar series
clean_solar_hist <- tsclean(ts_energy[,1])

# Plot solar series
autoplot(ts_energy[,1],series="Original") +
  autolayer(clean_solar_hist,series="Cleaned") +
  labs(title = "Outliers in Solar Energy Series [With Historical Data]")
```




```r
# Apply tsclean function for wind series
clean_wind_hist <- tsclean(ts_energy[,2])

# Plot wind series
autoplot(ts_energy[,2],series="Random") +
  autolayer(clean_wind_hist,series="Cleaned") +
  labs(title = "Outliers in Wind Energy Series [With Historical Data]")
```

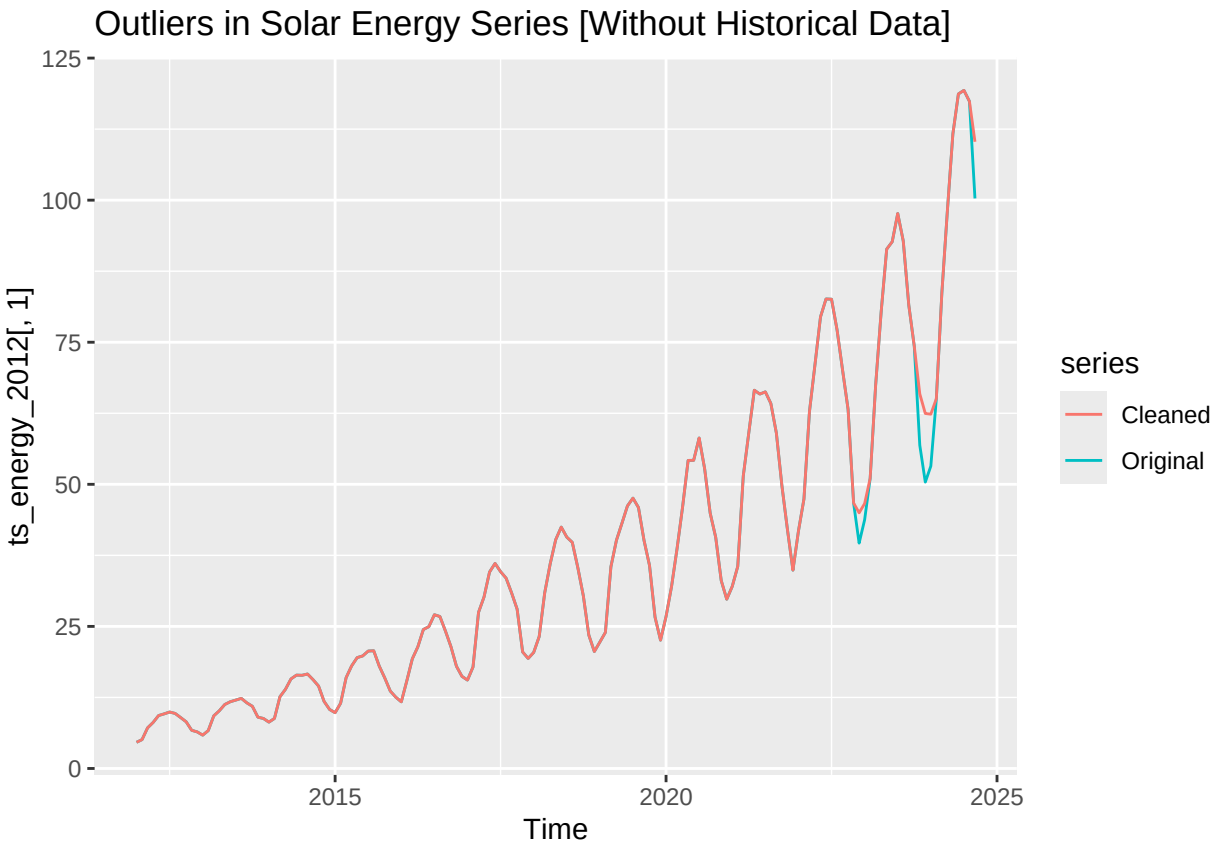


Answer: When the 'tsclean()' function is used to remove outliers in both data series inclusive of the historical data, the function considers all the spikes due to increasing trends as outliers and removes all of them. Removal of these data points might lead to misleading inaccurate results in time series analysis and forecasting.

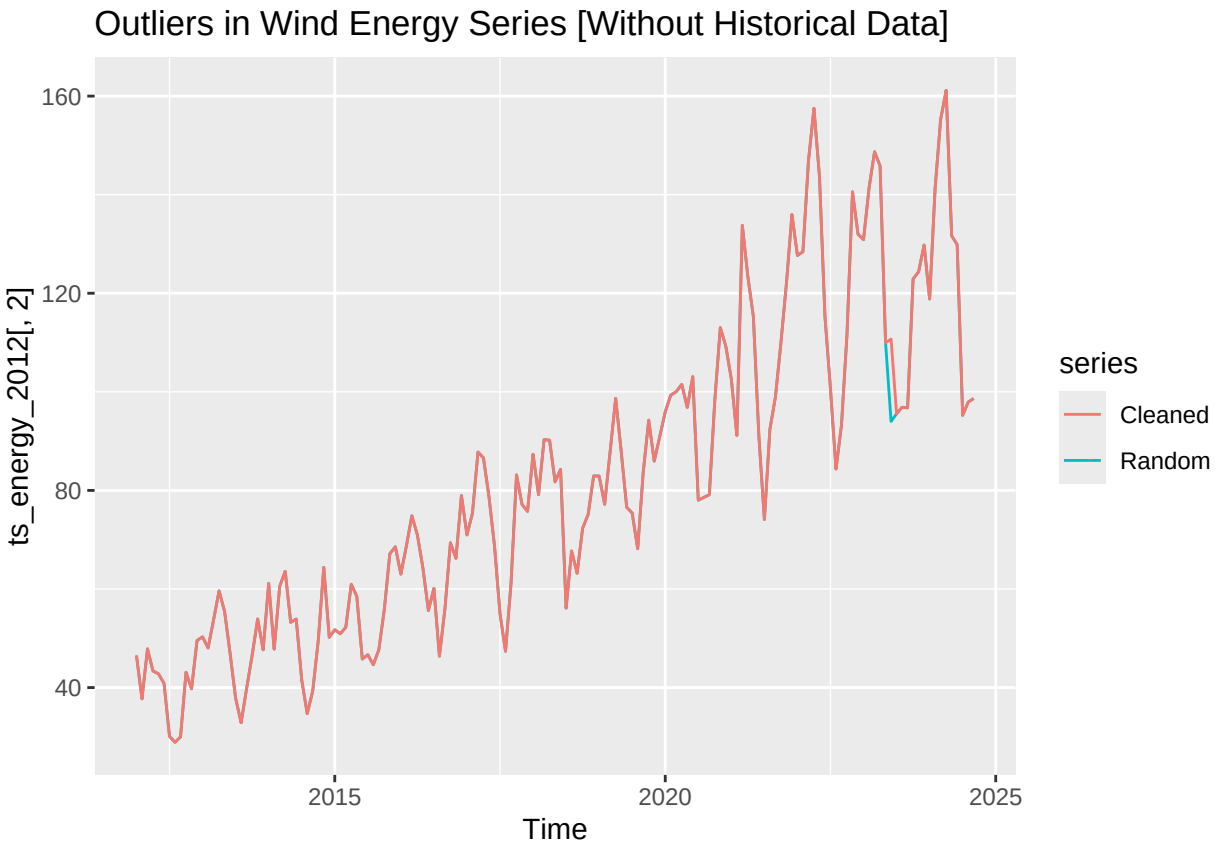
Q9

Redo number Q8 but now with the time series you created on Q7, i.e., the series starting in 2012. Using what autoplot() again what happened now? Did the function removed any outliers from the series?

```
# Apply tsclean function for solar series
clean_solar_2012 <- tsclean(ts_energy_2012[,1])

# Plot solar series
autoplot(ts_energy_2012[,1],series="Original") +
  autolayer(clean_solar_2012,series="Cleaned") +
  labs(title = "Outliers in Solar Energy Series [Without Historical Data]")
```




```r
# Apply tsclean function for wind series
clean_wind_2012 <- tsclean(ts_energy_2012[,2])

# Plot wind series
autoplot(ts_energy_2012[,2],series="Random") +
  autolayer(clean_wind_2012,series="Cleaned") +
  labs(title = "Outliers in Wind Energy Series [Without Historical Data]")
```



Answer: Compared to the Question 7, removing outliers in Question 8 become more accurate. The risk of blindly taking out the spikes existing due to upward trends was reduced in Question 8. However, in the solar energy series, the outliers removed between 2023 and 2025 might not be needed. It should be considered adding back again depending on the forecast that the study would like to make.