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

Assignment 5 - Due date 02/28/22

Sarah Mansfield

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.\

```
#Load/install required package here
library(readxl)
library(forecast)
```

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

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

```
##
## 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 1.3.1 --
```

```
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- 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()
```

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 January 2021
Monthly Energy Review.

```
energy_data <- read_excel("~/ENVIRON 790/ENV790_TimeSeriesAnalysis_Sp2022/Data/Table_10.1_Renewable_Ene
                      skip = 10)

energy_data <- energy_data %>%
  slice(2:nrow(energy_data))

head(energy_data)
```

```
## # A tibble: 6 x 14
##   Month               `Wood Energy Production` `Biofuels Produ~ `Total Biomass ~
##   <dttm>              <chr>                    <chr>            <chr>
## 1 1973-01-01 00:00:00 129.63                   Not Available    129.787
## 2 1973-02-01 00:00:00 117.194                  Not Available    117.338
## 3 1973-03-01 00:00:00 129.763                  Not Available    129.938
## 4 1973-04-01 00:00:00 125.462                  Not Available    125.636
## 5 1973-05-01 00:00:00 129.624                  Not Available    129.834
## 6 1973-06-01 00:00:00 125.435                  Not Available    125.611
## # ... with 10 more variables: Total Renewable Energy Production <chr>,
## #   Hydroelectric Power Consumption <chr>, Geothermal Energy Consumption <chr>,
## #   Solar Energy Consumption <chr>, Wind Energy Consumption <chr>,
## #   Wood Energy Consumption <chr>, Waste Energy Consumption <chr>,
## #   Biofuels Consumption <chr>, Total Biomass Energy Consumption <chr>,
## #   Total Renewable Energy Consumption <chr>
```

```
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!

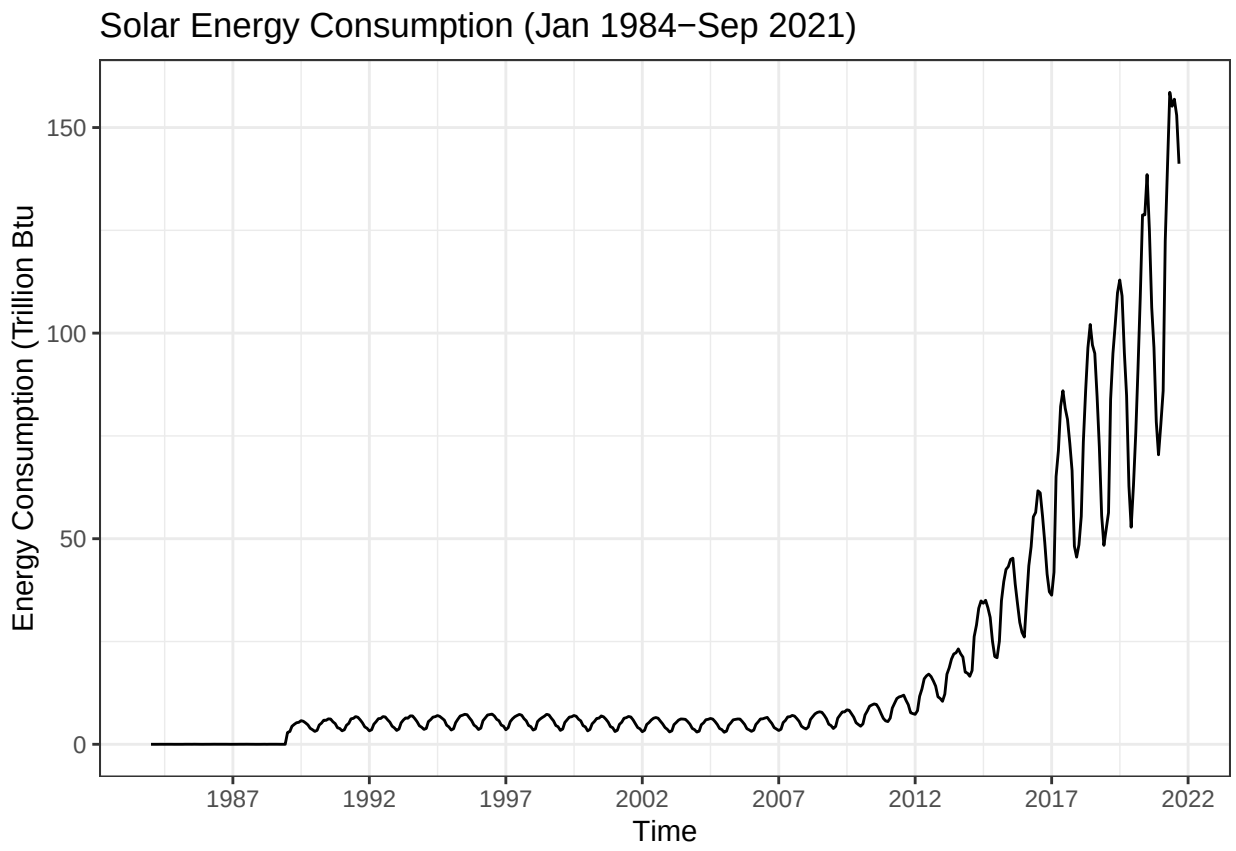
```
energy_data <- energy_data %>%
  select(Month, `Solar Energy Consumption`, `Wind Energy Consumption`) %>%
  mutate(Month = as.Date(Month),
         `Solar Energy Consumption` = suppressWarnings(as.numeric(`Solar Energy Consumption`)),
         `Wind Energy Consumption` = suppressWarnings(as.numeric(`Wind Energy Consumption`))) %>%
  drop_na()
head(energy_data)
```

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

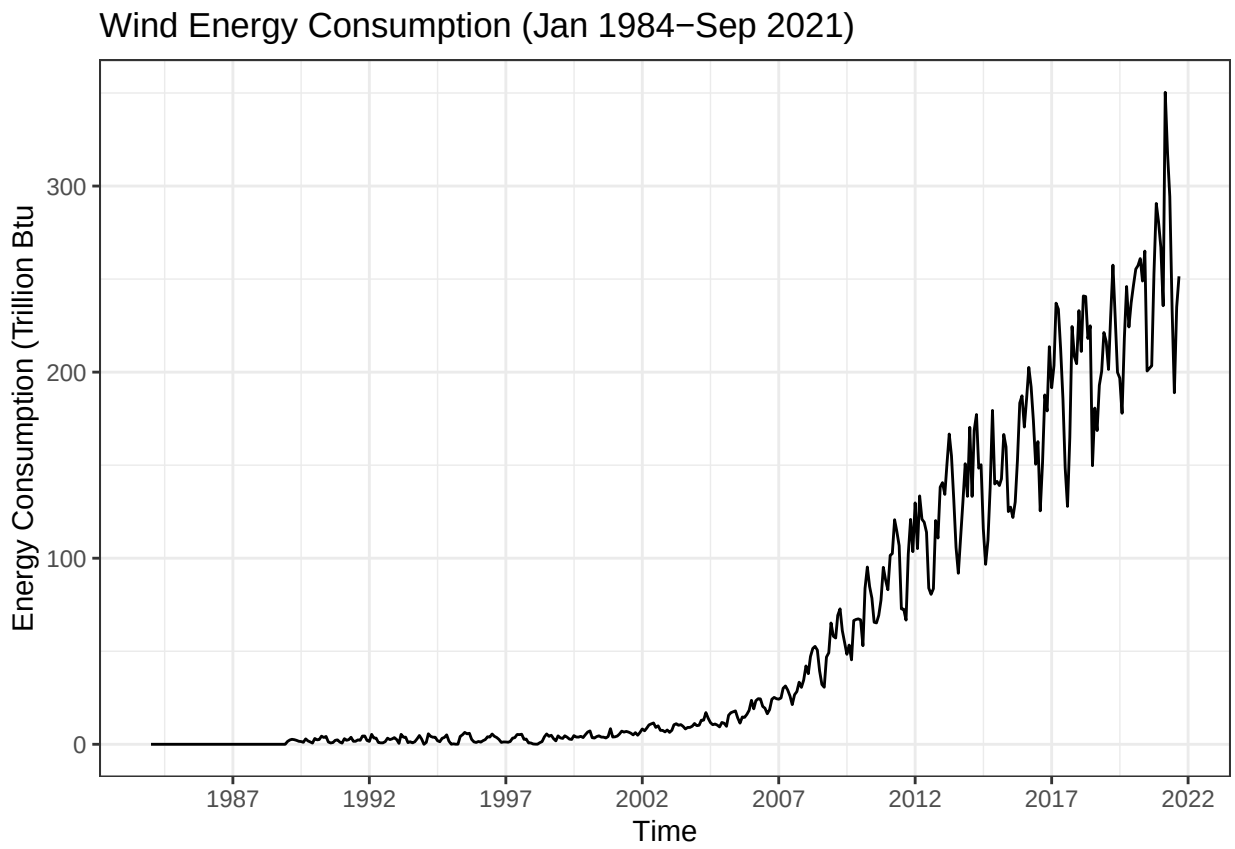
Q2

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")")*

```
ggplot(energy_data, aes(x = Month, y = `Solar Energy Consumption`)) +
  geom_line() +
  labs(title = "Solar Energy Consumption (Jan 1984-Sep 2021)",
  x = "Time", y = "Energy Consumption (Trillion Btu") +
  scale_x_date(date_breaks = "5 years", date_labels = "%Y") +
  theme_bw()
```



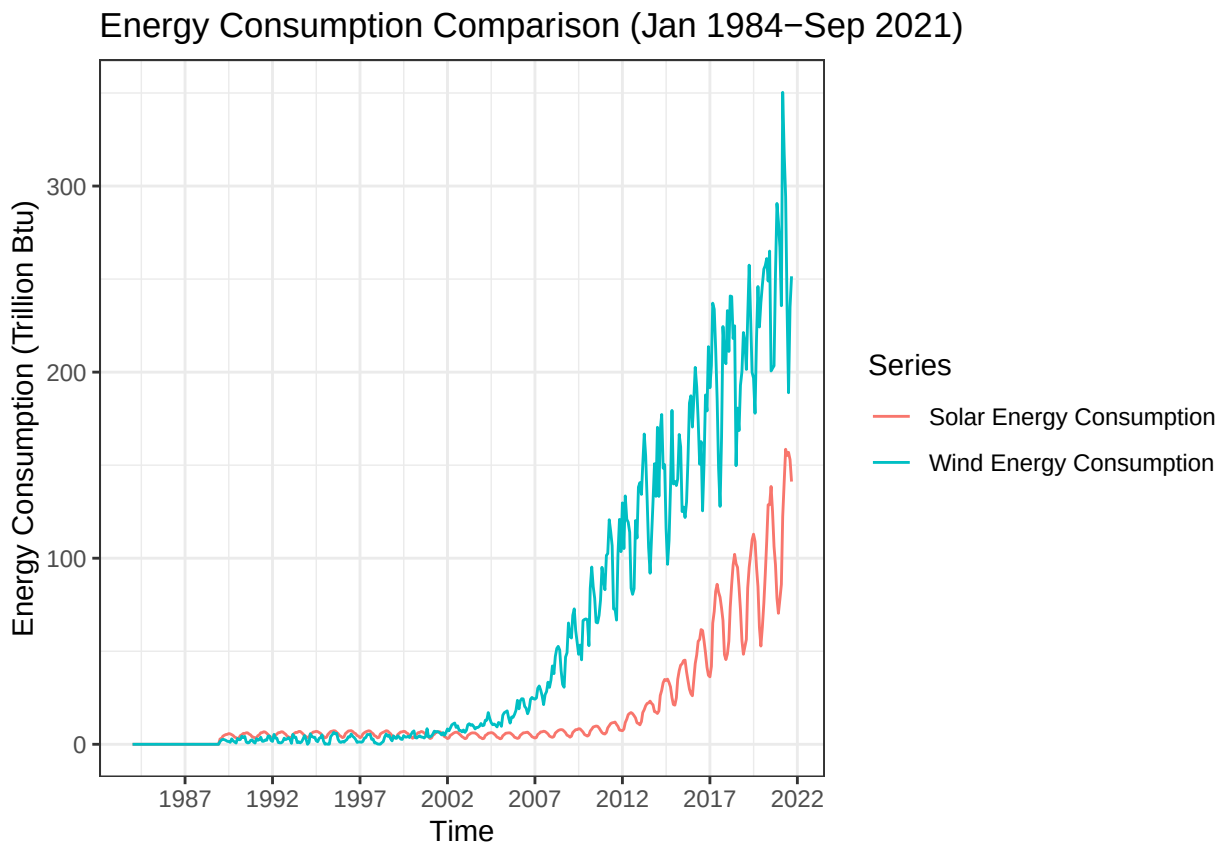

```
ggplot(energy_data, aes(x = Month, y = `Wind Energy Consumption`)) +
  geom_line() +
  labs(title = "Wind Energy Consumption (Jan 1984-Sep 2021)",
  x = "Time", y = "Energy Consumption (Trillion Btu") +
  scale_x_date(date_breaks = "5 years", date_labels = "%Y") +
  theme_bw()
```



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.

```
energy_data %>%
  pivot_longer(`Solar Energy Consumption`:`Wind Energy Consumption`,
               names_to = "series") %>%
  ggplot(aes(x = Month, y = value, color = series)) +
    geom_line() +
    labs(title = "Energy Consumption Comparison (Jan 1984-Sep 2021)",
         x = "Time", y = "Energy Consumption (Trillion Btu)", color = "Series") +
    scale_x_date(date_breaks = "5 years", date_labels = "%Y") +
    theme_bw()
```

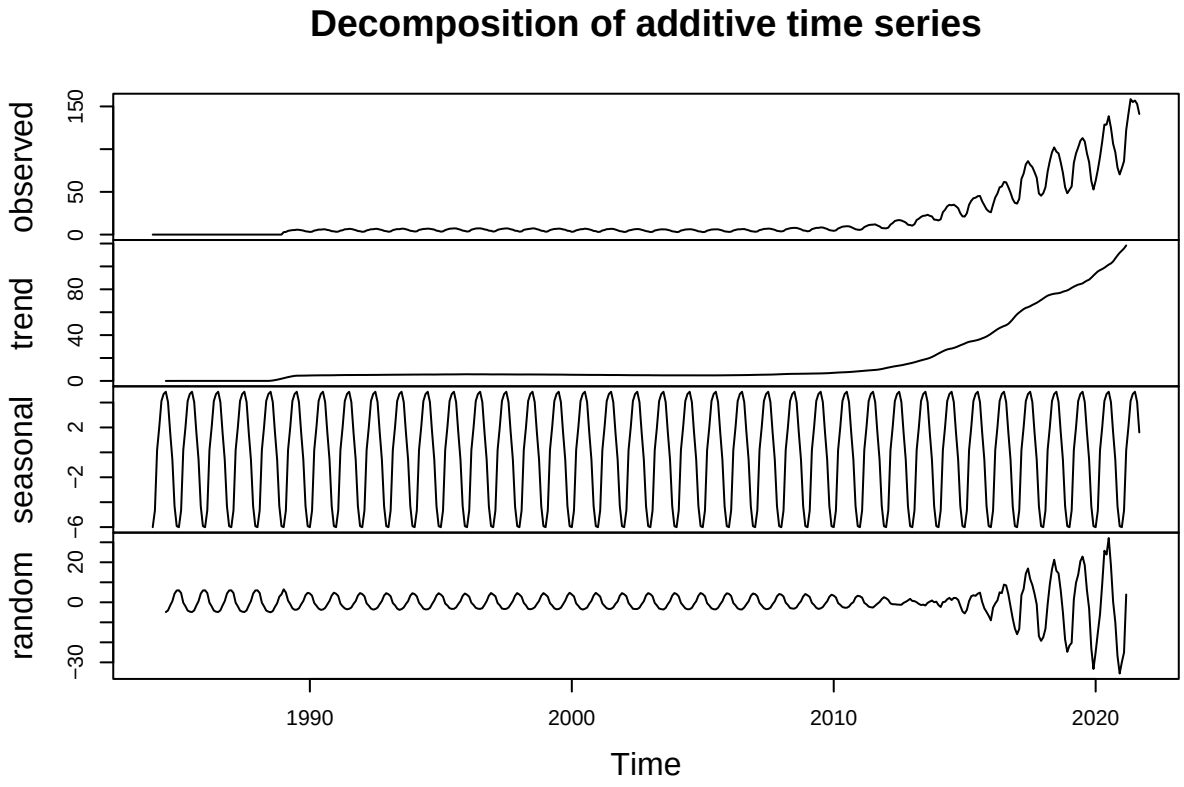


Q3

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?

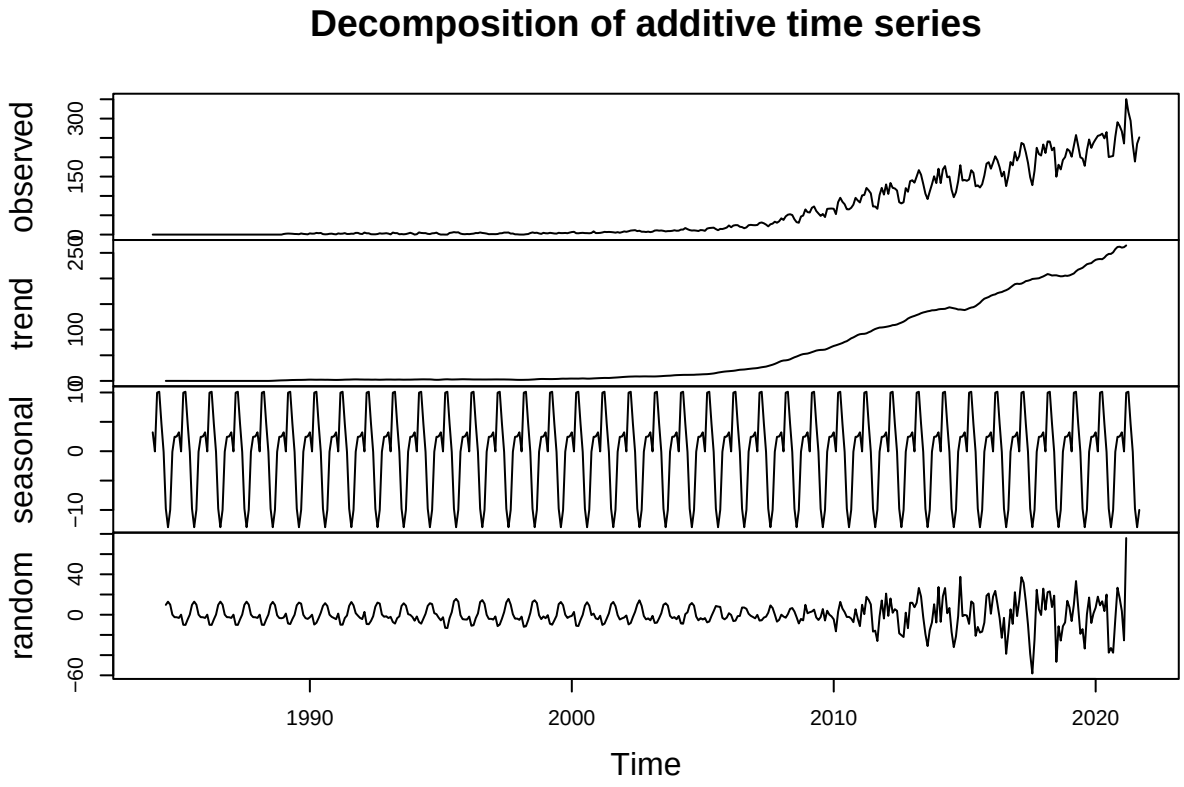
```
energy_ts <-  ts(energy_data[,2:3],
                 start = c(year(energy_data$Month[1]),
                           month(energy_data$Month[1])),
                 frequency = 12)
```

```
decompose_solar <- decompose(energy_ts[,"Solar Energy Consumption"], "additive")
plot(decompose_solar)
```



For the solar series, the trend component indicates that the data is nearly level (or has an extremely gradual increase) until 2010, after which there is a noticeable increasing trend in the data. The random component does still appear to have seasonality, as we can see a recurring pattern of consistent peaks and dips in the plot. Note that although the variance starts to increase around 2015, we still see recurring peaks and dips in the data.

```
decompose_wind <- decompose(energy_ts[,"Wind Energy Consumption"], "additive")
plot(decompose_wind)
```

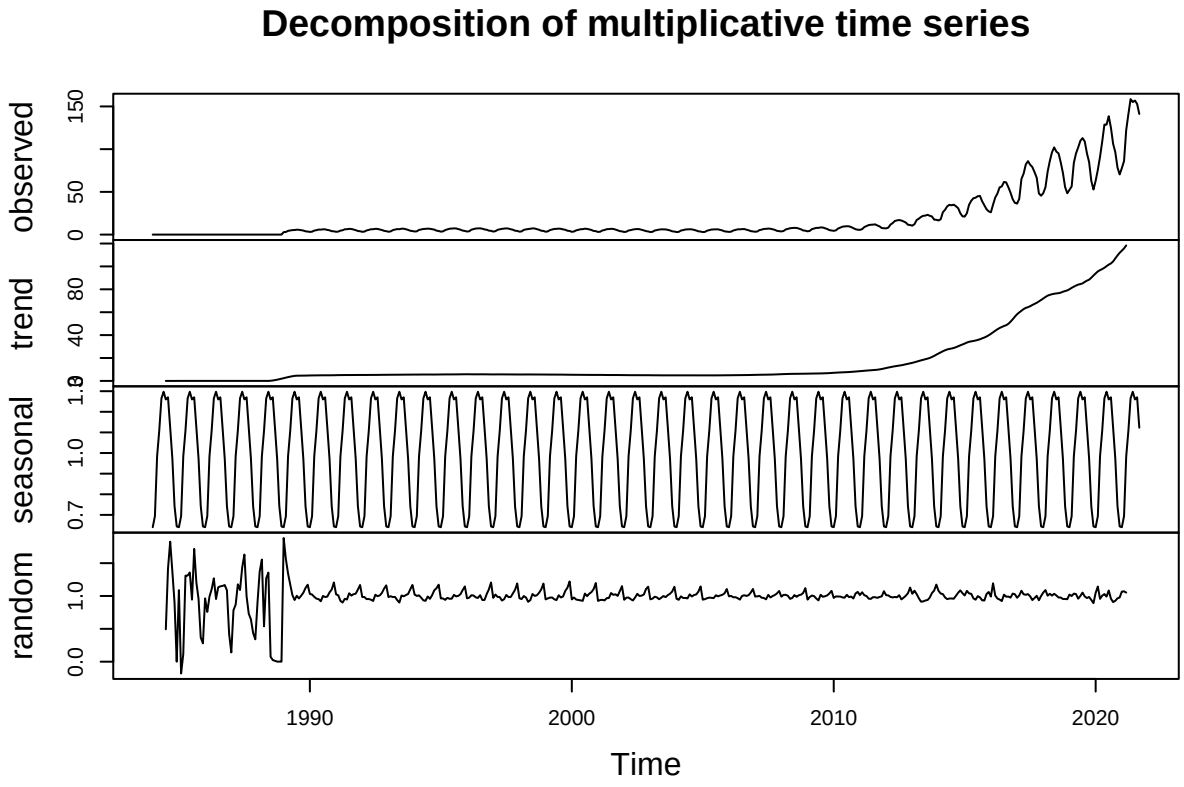


For the wind series, the trend component indicates that the data is pretty level up until 2000, starts to gradually increase from 2000 to around 2005, then starts to have a noticeable increasing trend in the data. Like the solar series plot, the random component does still appear to have some seasonality, as we can see a recurring pattern in the plot up until a little bit before 2010.

Q4

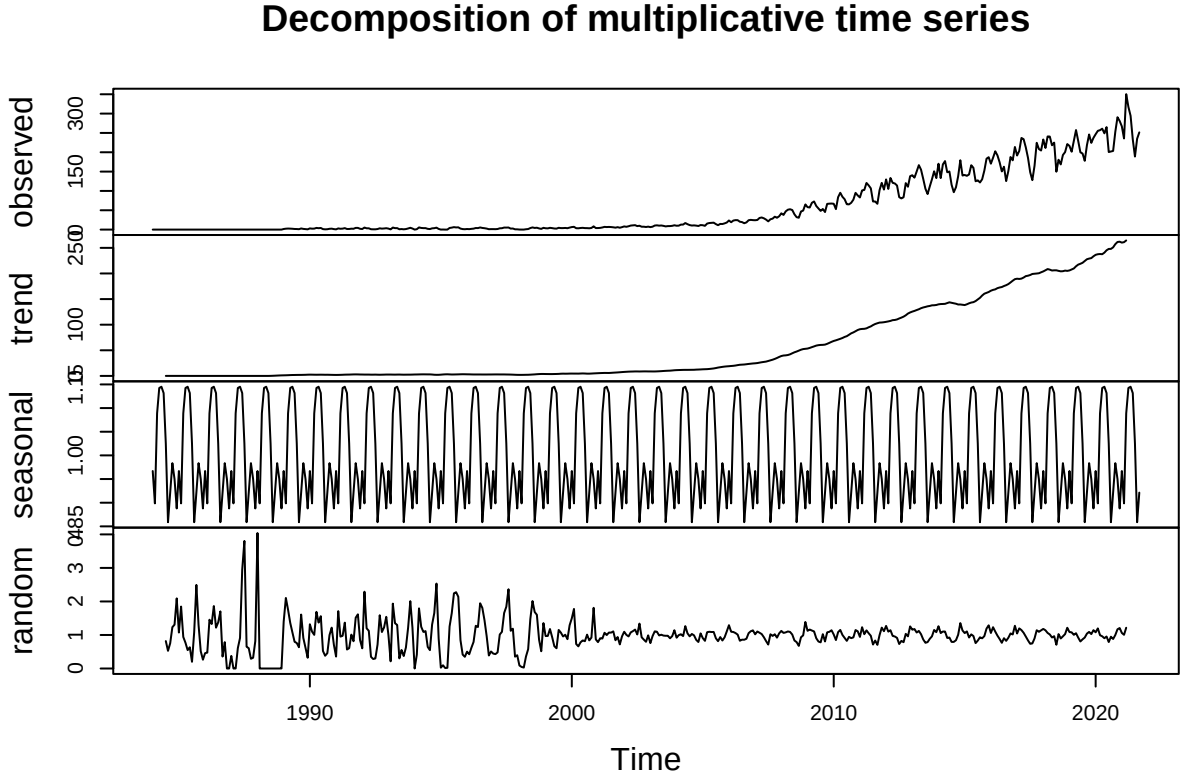
Use the decompose function again but now change the type of the seasonal component from additive to multiplicative. What happened to the random component this time?

```
decompose_solar <- decompose(energy_ts[,"Solar Energy Consumption"], "multiplicative")
plot(decompose_solar)
```



For the Solar series when we set the seasonal component to additive, we saw more fluctuations in the random component after 2015; on the other hand, when we set the seasonal component to multiplicative we see more fluctuations from the beginning until 1990.

```
decompose_wind <- decompose(energy_ts[,"Wind Energy Consumption"], "multiplicative")
plot(decompose_wind)
```



For the Wind series when we set the seasonal component to additive, we saw large fluctuations in the random component after 2010, and when we set the seasonal component to multiplicative we see that there are large fluctuations in the random component from the beginning until around 2000.

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: No, all of the historical data is likely not necessary since, as we've seen from the plots, energy consumption remained relatively low up until around 2010, after which there was a noticeable increase in energy consumption per month. As a result, using data from the 90s and early 20s wouldn't be very helpful for forecasting the next six months of energy consumption since data from those time periods remained relatively level and reflected a much different trend compared to the data from the last 10 years.

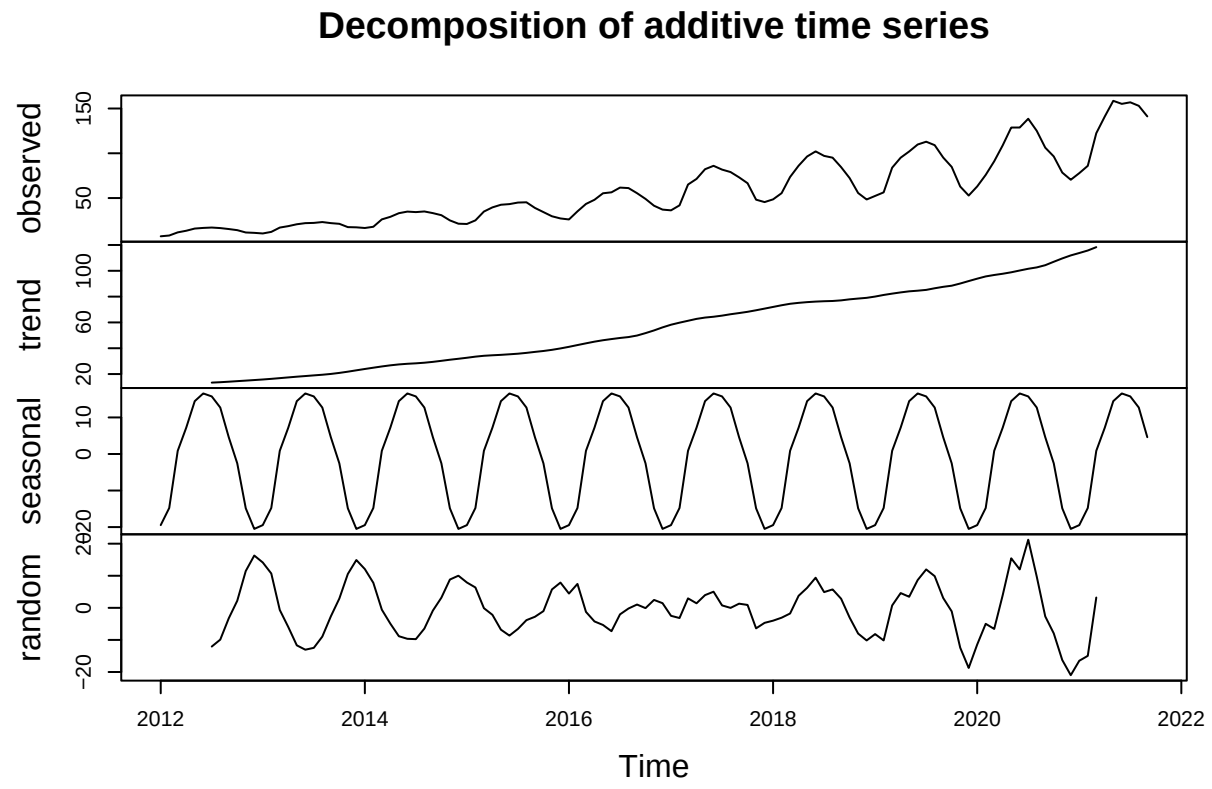
Q6

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 trying to remove the seasonal component and the challenge of trend on the seasonal component.

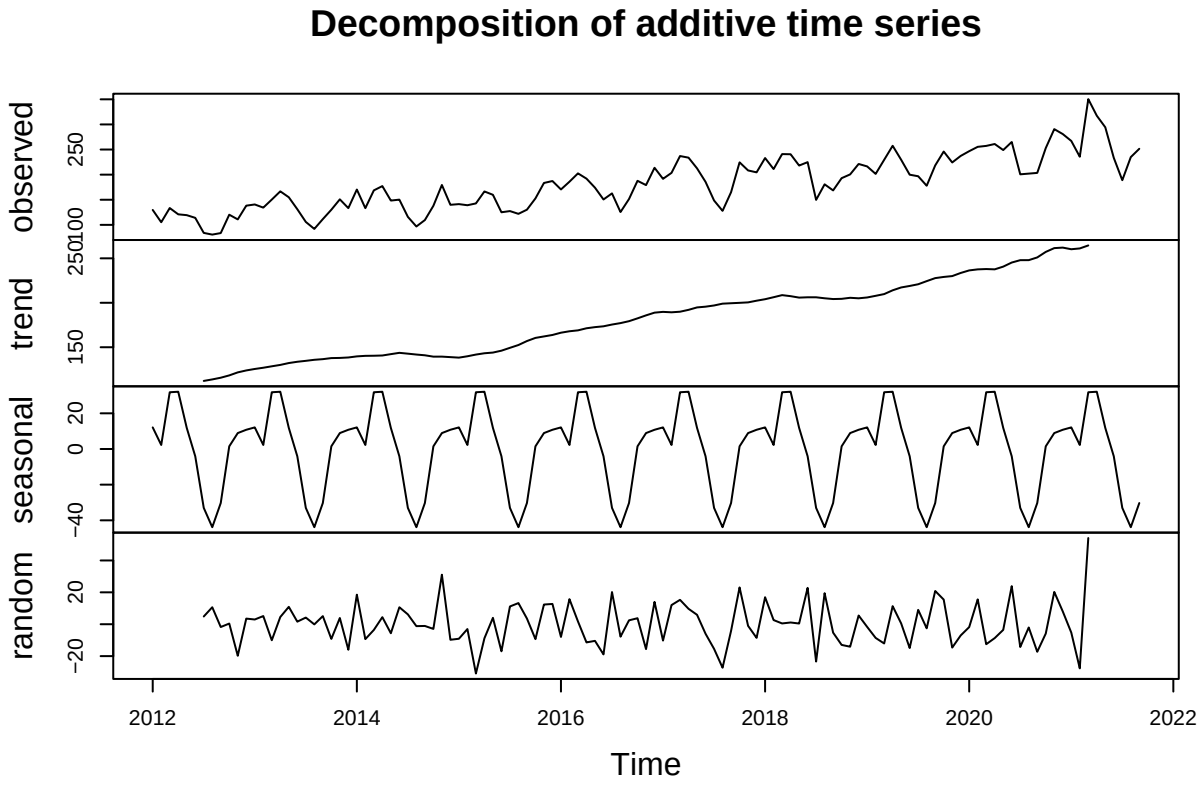
```r
energy_data <- energy_data %>%
  filter(year(Month) >= 2012)

energy_ts <-  ts(energy_data[,2:3],
                 start = c(year(energy_data$Month[1]),
                           month(energy_data$Month[1])),
                 frequency = 12)

decompose_solar <- decompose(energy_ts[,"Solar Energy Consumption"], "additive")
plot(decompose_solar)
```




```r
decompose_wind <- decompose(energy_ts[,"Wind Energy Consumption"], "additive")
plot(decompose_wind)
```



Answer: For both series, we can see that over time there's a pretty steady increasing trend in the data. For the solar series, the random component still seems to show some seasonality; on the other hand, the random component for the wind series appears more random. The challenge here is that as the overall trend increases for both series, so do the fluctuations in the seasonal component, which makes it more difficult to properly decompose each series using the additive model.