

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

Assignment 5 - Due date 02/28/22

Student Name

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(xlsx)
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.1.2
```

```
## 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 you can clean the data frame using pipes
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v tibble  3.1.4    v dplyr    1.0.7
```

```
## v tidyr   1.1.3    v stringr 1.4.0
```

```
## v readr   2.0.1    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_Consumption”. The data comes from the US Energy Information and Administration and corresponds to the January 2021 Monthly Energy Review.

```
#Importing data set - using xlsx package
energy_data <- read.xlsx(file="/Users/emreyurtbay/Documents/Duke/env790/ENV790_TimeSeriesAnalysis_Sp2021.xlsx")

#Now let's extract the column names from row 11 only
read_col_names <- read.xlsx(file="/Users/emreyurtbay/Documents/Duke/env790/ENV790_TimeSeriesAnalysis_Sp2021.xlsx", sheet = "Table_10.1_Renewable_Energy_Production_and_Consumption", col = 11)

colnames(energy_data) <- read_col_names
head(energy_data)
```

```
##      Month Wood Energy Production Biofuels Production
## 1 1973-01-01          129.630      Not Available
## 2 1973-02-01          117.194      Not Available
## 3 1973-03-01          129.763      Not Available
## 4 1973-04-01          125.462      Not Available
## 5 1973-05-01          129.624      Not Available
## 6 1973-06-01          125.435      Not Available
##      Total Biomass Energy Production Total Renewable Energy Production
## 1          129.787          403.981
## 2          117.338          360.900
## 3          129.938          400.161
## 4          125.636          380.470
## 5          129.834          392.141
## 6          125.611          377.232
##      Hydroelectric Power Consumption Geothermal Energy Consumption
## 1          272.703          1.491
## 2          242.199          1.363
## 3          268.810          1.412
## 4          253.185          1.649
## 5          260.770          1.537
## 6          249.859          1.763
##      Solar Energy Consumption Wind Energy Consumption Wood Energy Consumption
## 1      Not Available      Not Available          129.630
## 2      Not Available      Not Available          117.194
## 3      Not Available      Not Available          129.763
## 4      Not Available      Not Available          125.462
## 5      Not Available      Not Available          129.624
## 6      Not Available      Not Available          125.435
##      Waste Energy Consumption Biofuels Consumption
## 1          0.157      Not Available
```

```
## 2          0.144      Not Available
## 3          0.176      Not Available
## 4          0.174      Not Available
## 5          0.210      Not Available
## 6          0.176      Not Available
##   Total Biomass Energy Consumption Total Renewable Energy Consumption
## 1                129.787                403.981
## 2                117.338                360.900
## 3                129.938                400.161
## 4                125.636                380.470
## 5                129.834                392.141
## 6                125.611                377.232
```

```
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$`Solar Energy Consumption` <- as.numeric(energy_data$`Solar Energy Consumption`)

## Warning: NAs introduced by coercion

energy_data$`Wind Energy Consumption` <- as.numeric(energy_data$`Wind Energy Consumption`)

## Warning: NAs introduced by coercion

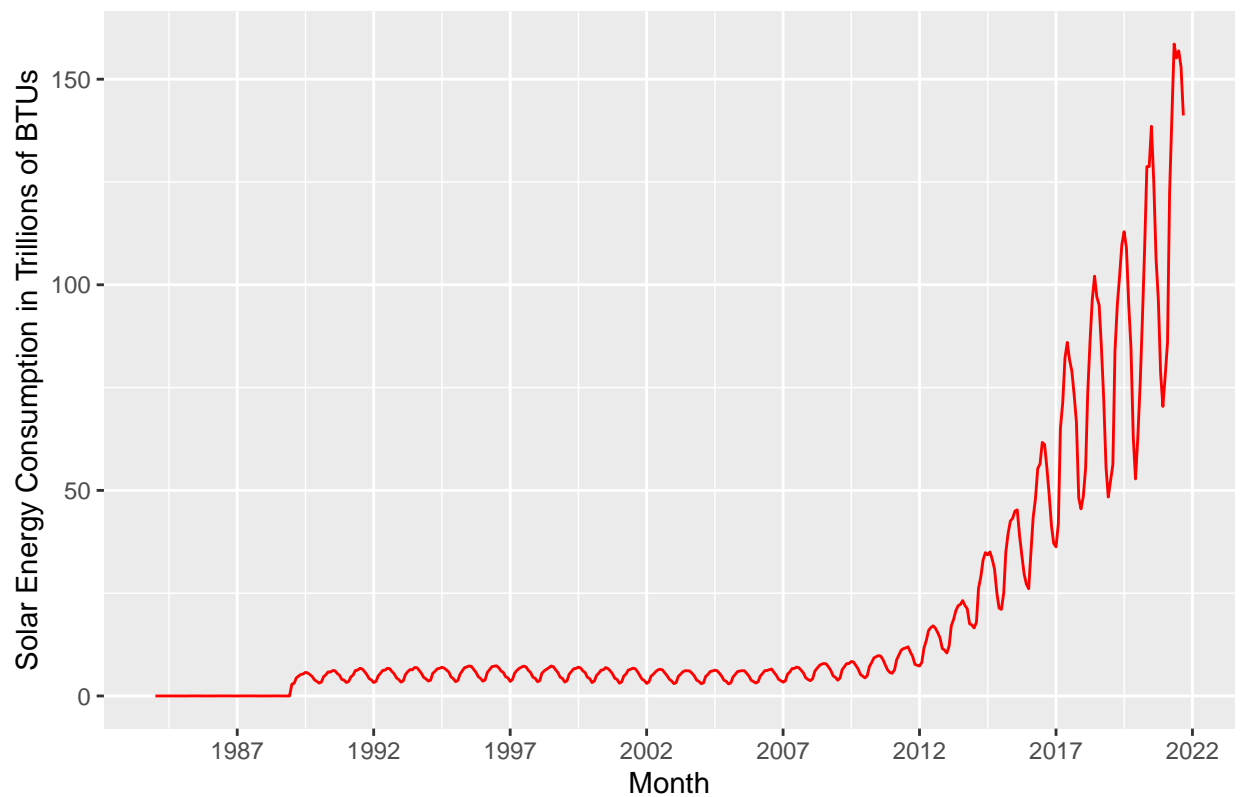
energy <- energy_data %>%
  select(Month, `Solar Energy Consumption`,
         `Wind Energy Consumption`) %>%
  drop_na()
```

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")`

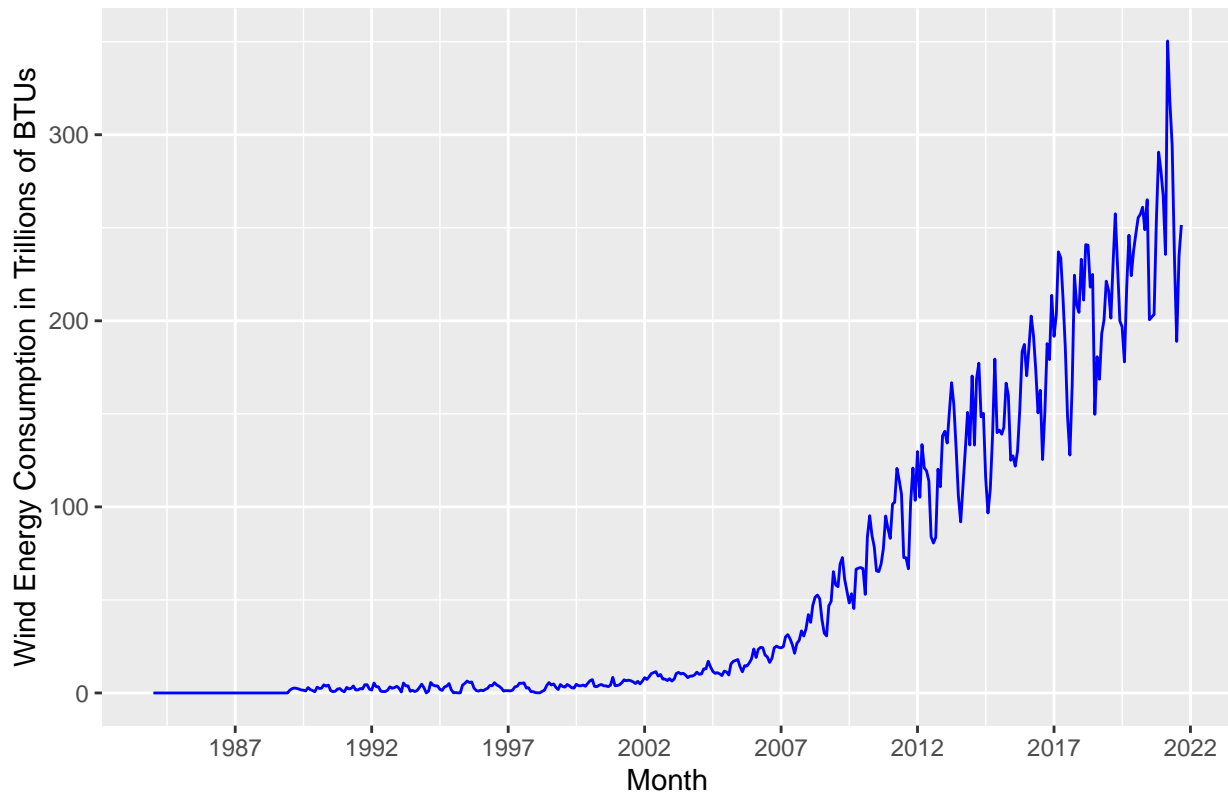
```
ggplot(energy, aes(Month, `Solar Energy Consumption`)) +
  geom_line(color = "red") +
  scale_x_date(date_breaks = "5 years", date_labels = "%Y") +
  ylab("Solar Energy Consumption in Trillions of BTUs") +
  ggtitle("Solar Energy Consumption over Time")
```

Solar Energy Consumption over Time



```
ggplot(energy, aes(Month, `Wind Energy Consumption`)) +  
  geom_line(color = "blue") +  
  scale_x_date(date_breaks = "5 years", date_labels = "%Y") +  
  ylab("Wind Energy Consumption in Trillions of BTUs") +  
  ggtitle("Wind Energy Consumption over Time")
```

Wind Energy Consumption over Time

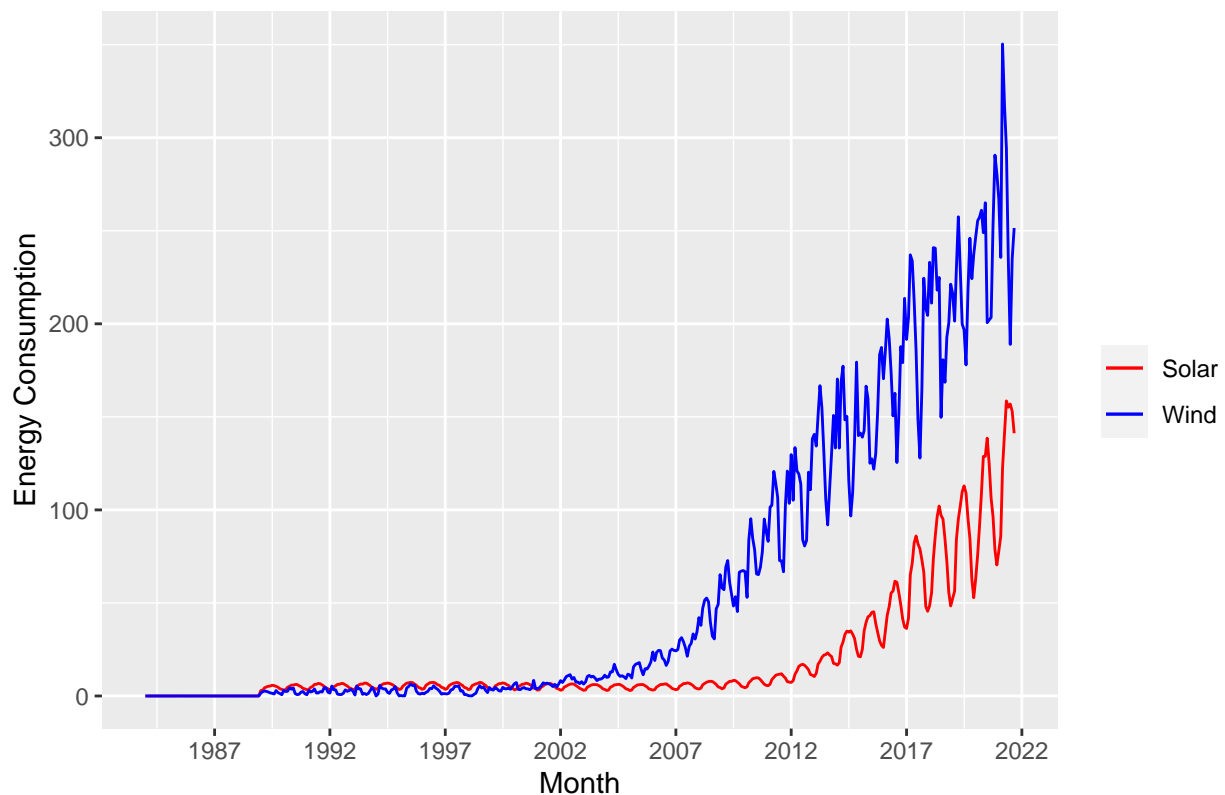


Q3

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) +  
  geom_line(aes(x = Month, y = `Solar Energy Consumption`, color = "Solar")) +  
  geom_line(aes(x = Month, y = `Wind Energy Consumption`, color = "Wind")) +  
  scale_color_manual("",  
                     breaks = c("Solar", "Wind"),  
                     values = c("red", "blue")) +  
  ylab("Energy Consumption")+  
  scale_x_date(date_breaks = "5 years", date_labels = "%Y") +  
  ggtitle("Solar and Wind Energy Consumption over Time")
```

Solar and Wind Energy Consumption over Time

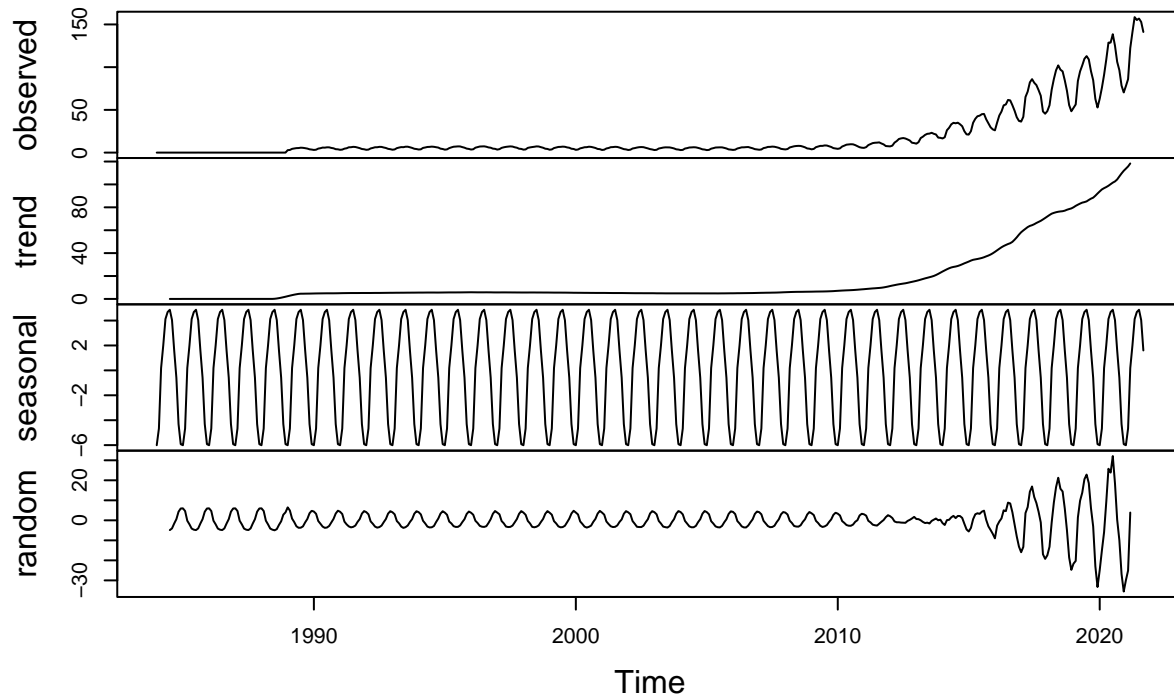


Q3

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?

```
solar_ts <- ts(energy$`Solar Energy Consumption`,  
              frequency = 12, start = c(1984, 1))  
wind_ts <- ts(energy$`Wind Energy Consumption`,  
             frequency = 12, start = c(1984, 1))  
  
solar_decomp <- decompose(solar_ts, type = "additive")  
wind_decomp <- decompose(wind_ts, type = "additive")  
  
plot(solar_decomp)
```

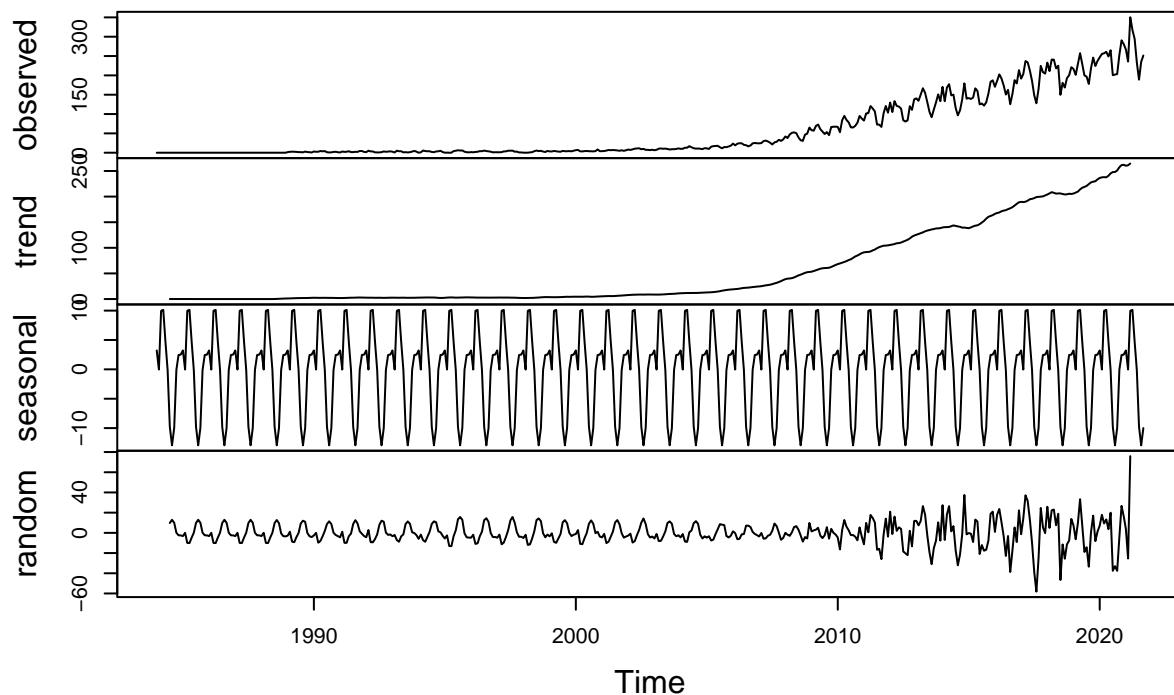
Decomposition of additive time series



For the solar data, we see very little upward or downward trend until about 2010, when we start to see an upward trend. The random component does not appear to be random - in fact, we seem to see some rather strong seasonality in the random component.

```
plot(wind_decomp)
```

Decomposition of additive time series



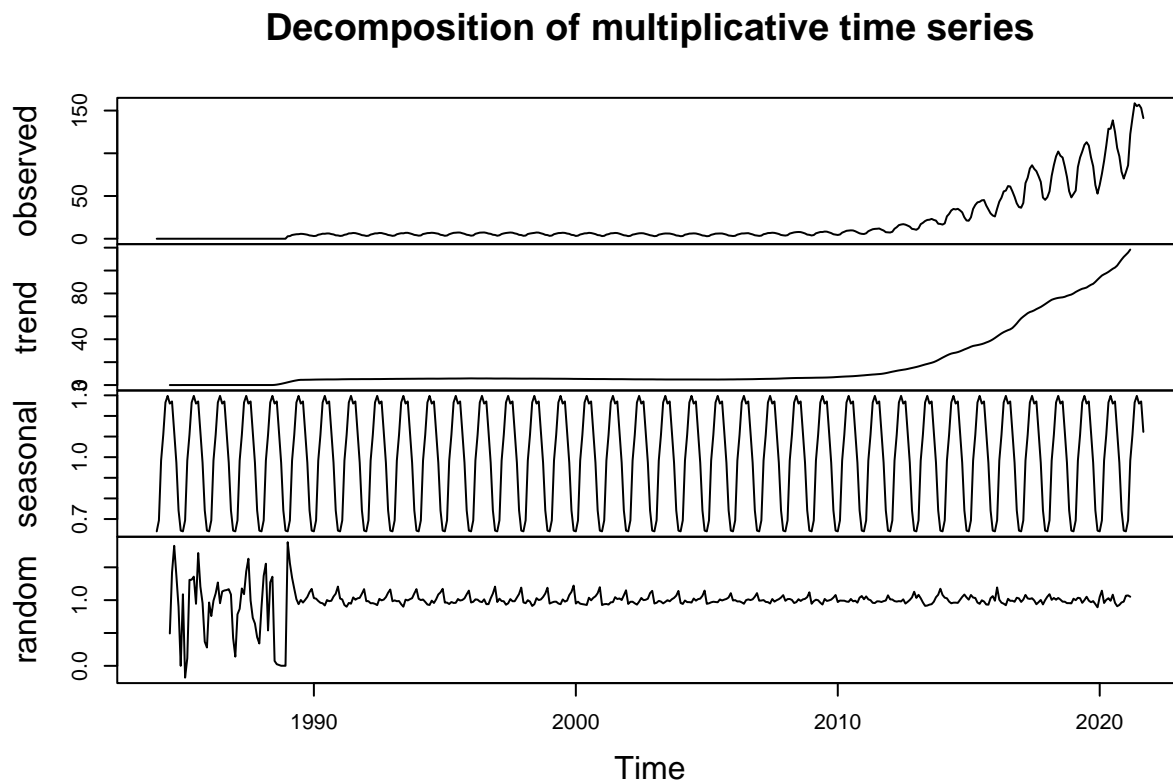
For the wind data, we see very little upward or downward trend until about 2000, when we start to see an upward trend. Again, the random component does not appear to be random - in fact, we seem to see some rather strong seasonality in the random component.

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?

```
solar_decomp_m <- decompose(solar_ts, type = "multiplicative")  
wind_decomp_m <- decompose(wind_ts, type = "multiplicative")
```

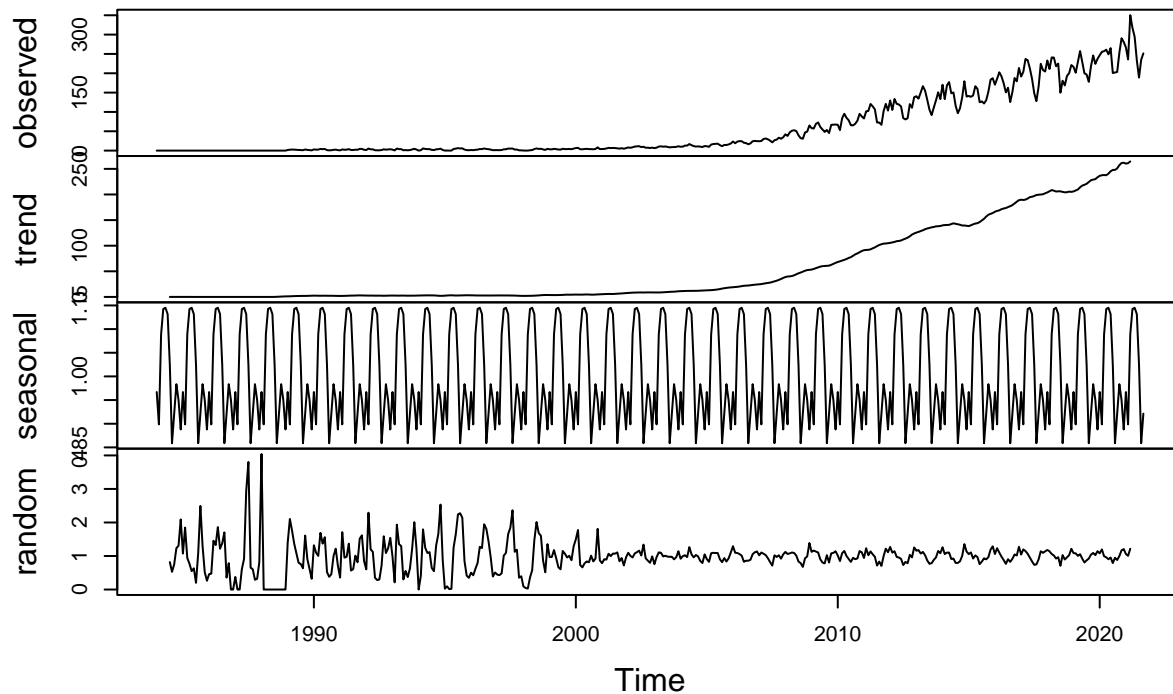
```
plot(solar_decomp_m)
```



For the solar data, we see more random behavior in the beginning of the random component of the series, and the seasonality is not as smooth as before, but we still see peaks and troughs with some regularity after performing the multiplicative decomposition. The random component looks more random, but not completely random.

```
plot(wind_decomp_m)
```


Decomposition of multiplicative time series



For the wind data, we see more random behavior in the beginning of the random component of the series, and the seasonality is less regular than before after performing the multiplicative decomposition. Toward the end of the random component series, we see more seasonal looking behavior, with regular peaks and troughs.

In general, the multiplicative model is appropriate if the seasonal fluctuations increase or decrease proportionally with increases and decreases in the level of the series, and this is what we see here, so I think the multiplicative model is more appropriate.

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: When trying to predict both series, we should not use all of the historical data. In both plots, we see very different behavior in different parts of the series. For both plots, we see consistently low consumption until about 2002. In 2002, the wind series starts to be trending up, and we see both seasonality and increasing trend until the present. To model the wind data, we shouldn't use data from at least before 2002 to predict past 2022. In about 2012, the solar series starts to be trending up, and we see both seasonality and increasing trend until the present. To model the solar data, we shouldn't use data from before 2012 to predict past 2022.

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(yyyy, 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.

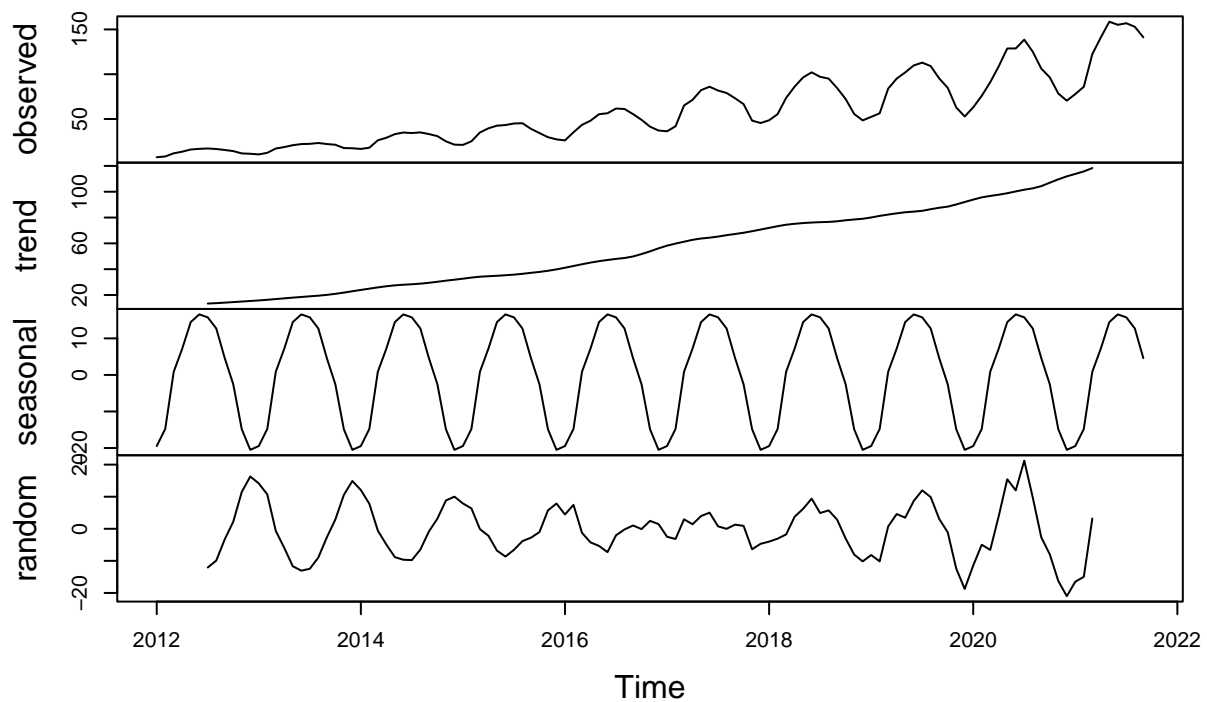
```
energy2012 <- energy %>%
  filter(year(Month) >= 2012)
```

```
solar_ts_2012 <- ts(energy2012$`Solar Energy Consumption`,
  frequency = 12, start = c(2012, 1))
wind_ts_2012 <- ts(energy2012$`Wind Energy Consumption`,
  frequency = 12, start = c(2012, 1))
```

```
solar_decomp_2012 <- decompose(solar_ts_2012, type = "additive")
wind_decomp_2012 <- decompose(wind_ts_2012, type = "additive")
```

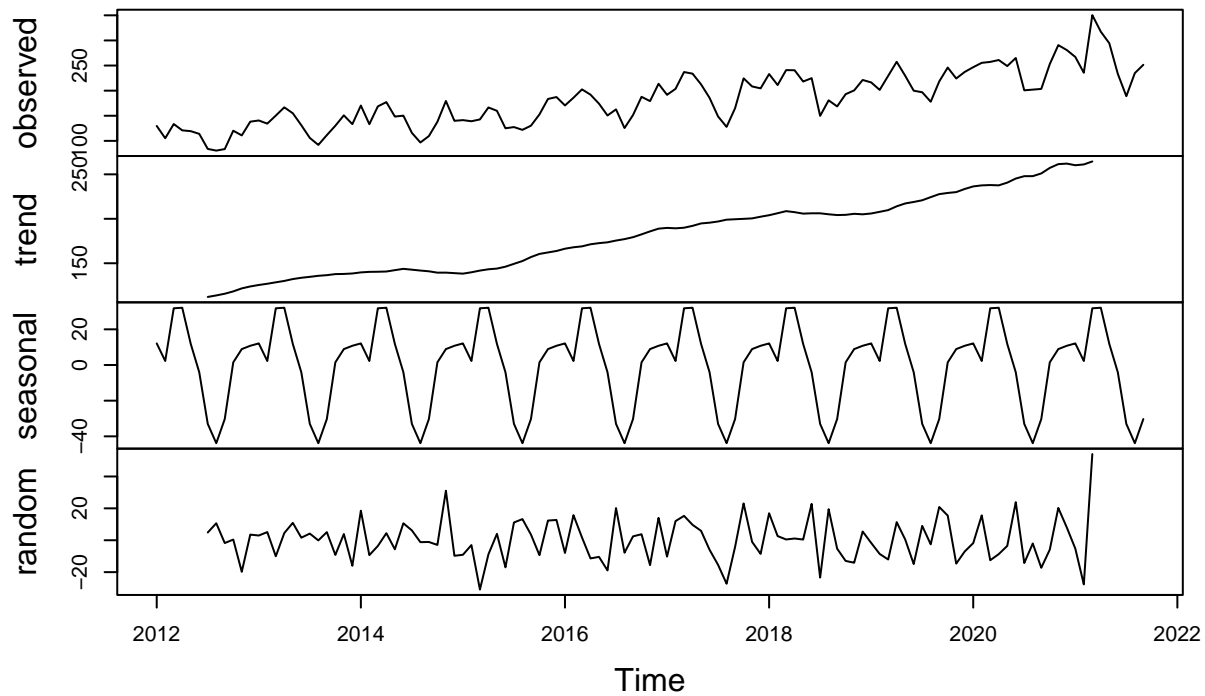
```
plot(solar_decomp_2012)
```

Decomposition of additive time series



```
plot(wind_decomp_2012)
```

Decomposition of additive time series



Answer: After the further transformations, we still see some seasonality in the random component of the time series for the solar data. We do not see the same behavior in the wind series - the random component looks more random. We have a challenge again because the seasonal fluctuations for both series increase proportionally with increases in the level of the series, which can make it hard for the additive model to properly decompose the series. Again, I think a multiplicative or log-additive model would be more appropriate here.