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

Assignment 5 - Due date 02/28/22

Aasha Reddy

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.\

```r
#Load/install required package here
library(xlsx)
```

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

```r
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
```

```r
library(tseries)
```

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

```r
library(ggplot2)
library(Kendall)
```

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

```r
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  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     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_Consumptio
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/Aasha Reddy/Documents/Statistics - Duke University/2022 Spring/Tin

#Now let's extract the column names from row 11 only
read_col_names <- read.xlsx(file="/Users/Aasha Reddy/Documents/Statistics - Duke University/2022 Spring/

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!

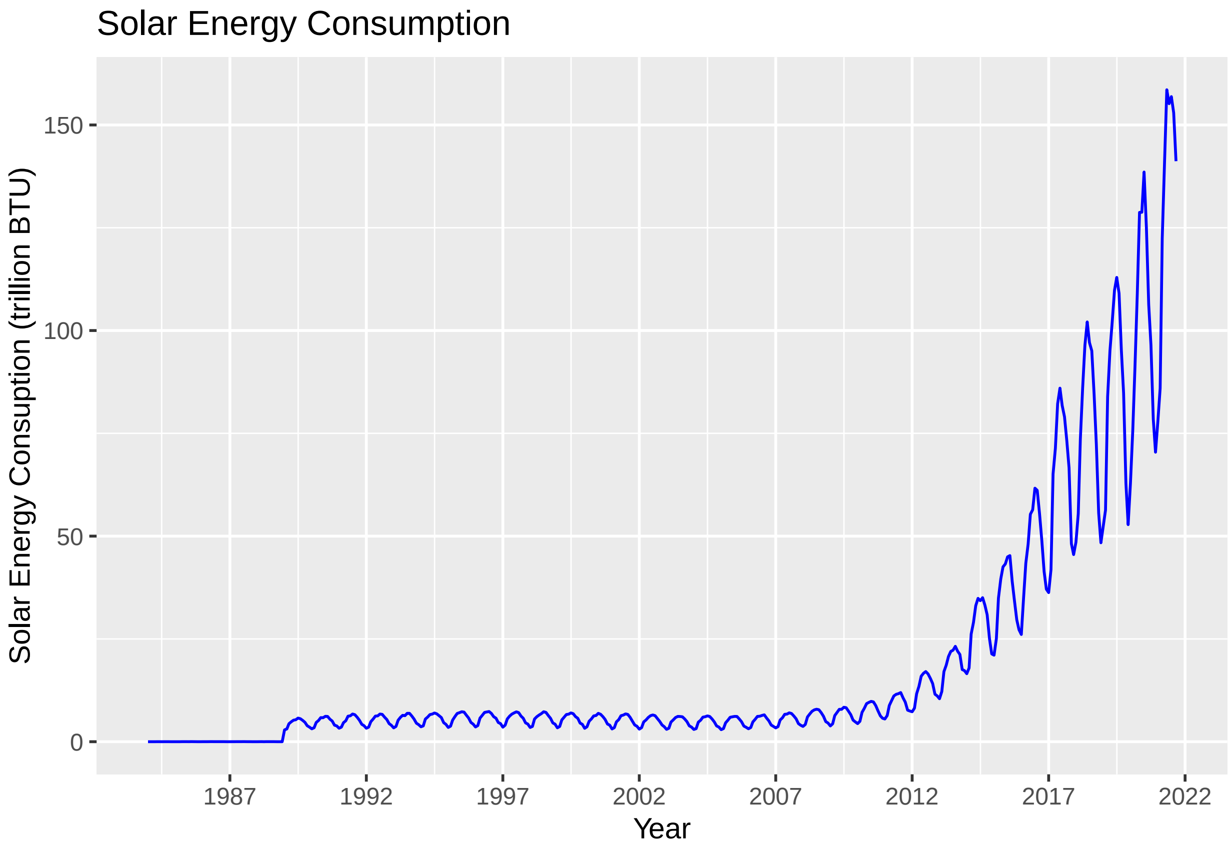
```
# change Not Available to NA
energy_data$`Solar Energy Consumption` <- ifelse(energy_data$`Solar Energy Consumption` == "Not Availabl
                                                 NA, energy_data$`Solar Energy Consumption`)

energy_data$`Wind Energy Consumption` <- ifelse(energy_data$`Wind Energy Consumption` == "Not Available
                                                NA, energy_data$`Wind Energy Consumption`)

energy_raw <- energy_data %>%
  select(Month, `Solar Energy Consumption`, `Wind Energy Consumption`) %>%
  mutate(`Solar Energy Consumption` = as.numeric(`Solar Energy Consumption`),
         `Wind Energy Consumption` = as.numeric(`Wind Energy Consumption`)) %>%
  mutate(Month = ymd(Month))

energy_raw <- drop_na(energy_raw)
```

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")")*

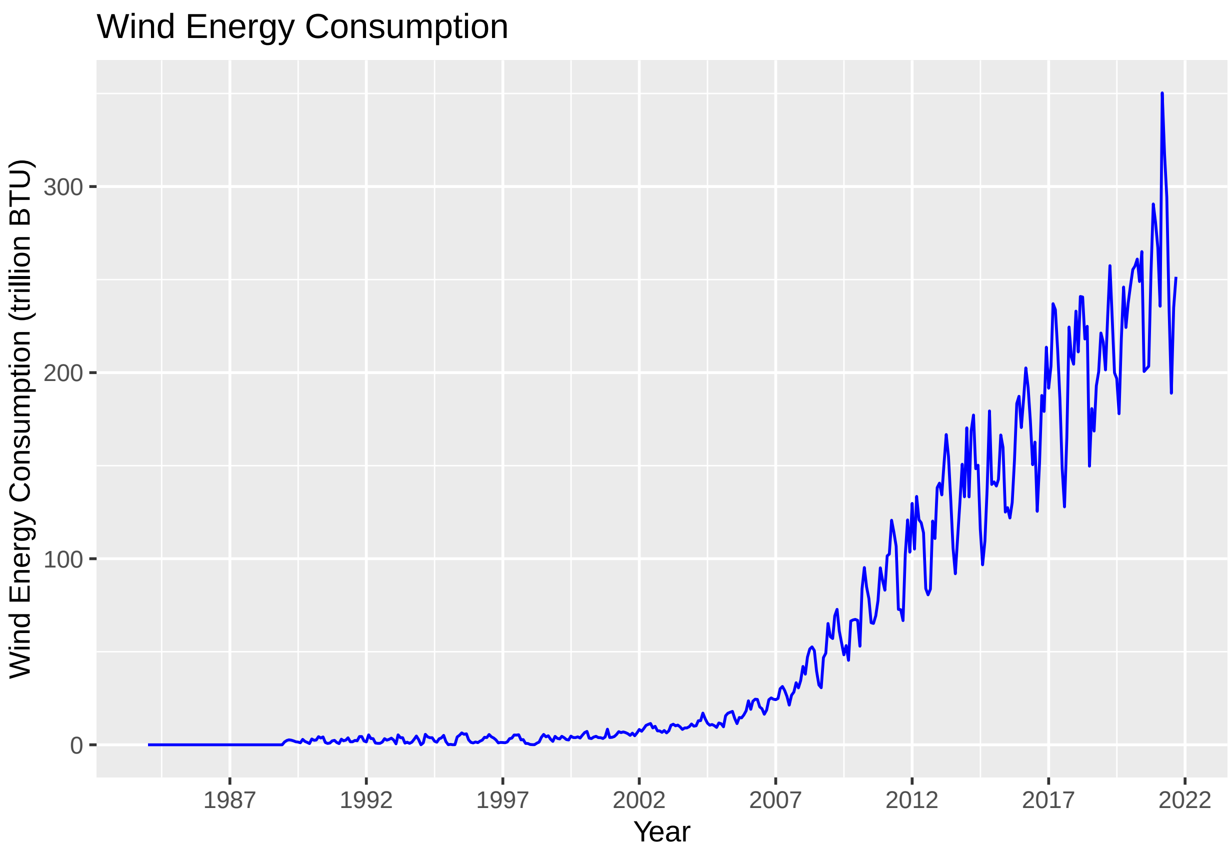
Solar Energy Consumption:

```
ggplot(energy_raw, aes(x = Month, y = `Solar Energy Consumption`)) +
            geom_line(color="blue") +
  labs(title = "Solar Energy Consumption",
       y = "Solar Energy Consuption (trillion BTU)",
       x = "Year") +
  scale_x_date(date_breaks = "5 years", date_labels = "%Y")
```



Wind Energy Consumption:

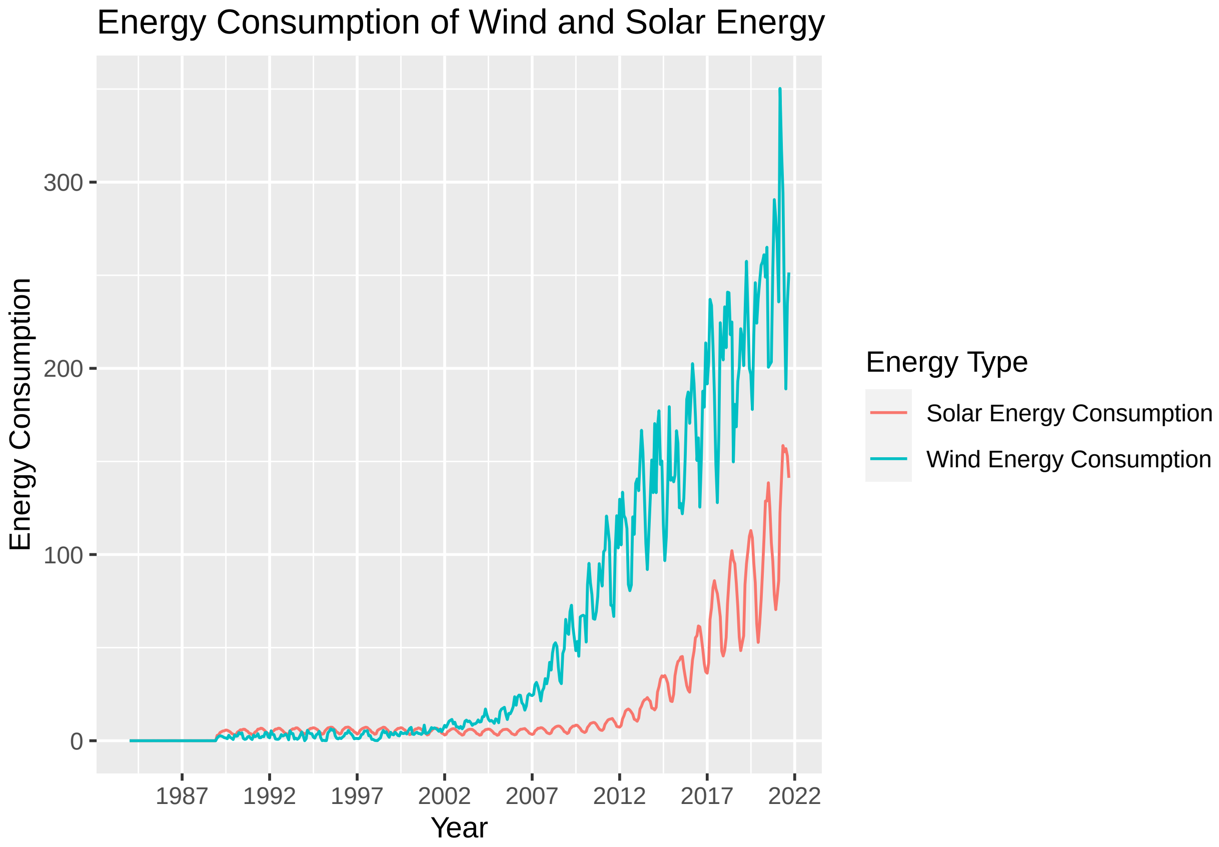
```
ggplot(energy_raw, aes(x = Month, y = `Wind Energy Consumption`)) +
            geom_line(color="blue") +
  labs(title = "Wind Energy Consumption",
    y = "Wind Energy Consumption (trillion BTU)",
    x = "Year") +
    scale_x_date(date_breaks = "5 years", date_labels = "%Y")
```



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.

```
energy_raw %>%
  pivot_longer(cols = 2:3, names_to = "Energy Type") %>%
  ggplot(., aes(x = Month, y = value, col = `Energy Type`)) +
  geom_line() +
  labs(title = "Energy Consumption of Wind and Solar Energy",
       x = "Year") +
  ylab("Energy Consumption") +
  scale_x_date(date_breaks = "5 years", date_labels = "%Y")
```



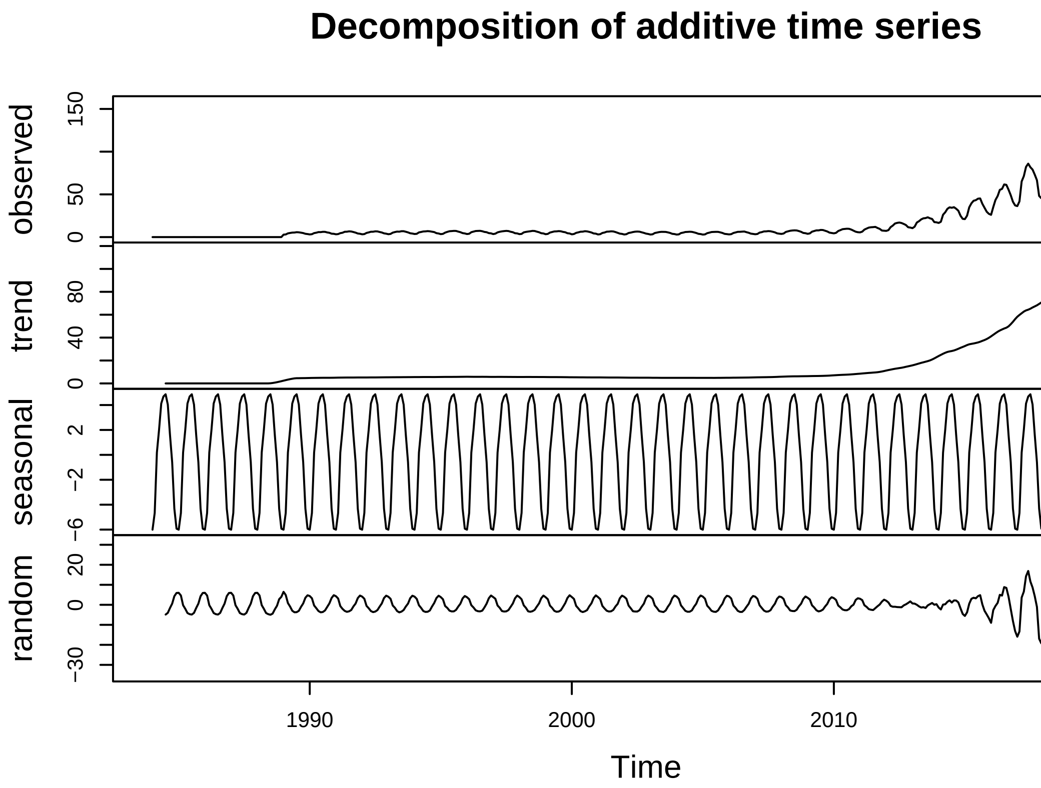
Q3

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?

```
# transform data into time series object
energy_ts <- ts(data = energy_raw %>% select(-Month), start = c(1984, 1), frequency = 12)
```

```
solar_decomp <- decompose(energy_ts[,1], type = "additive")
```

```
# plot trend and random componenet
plot(solar_decomp)
```

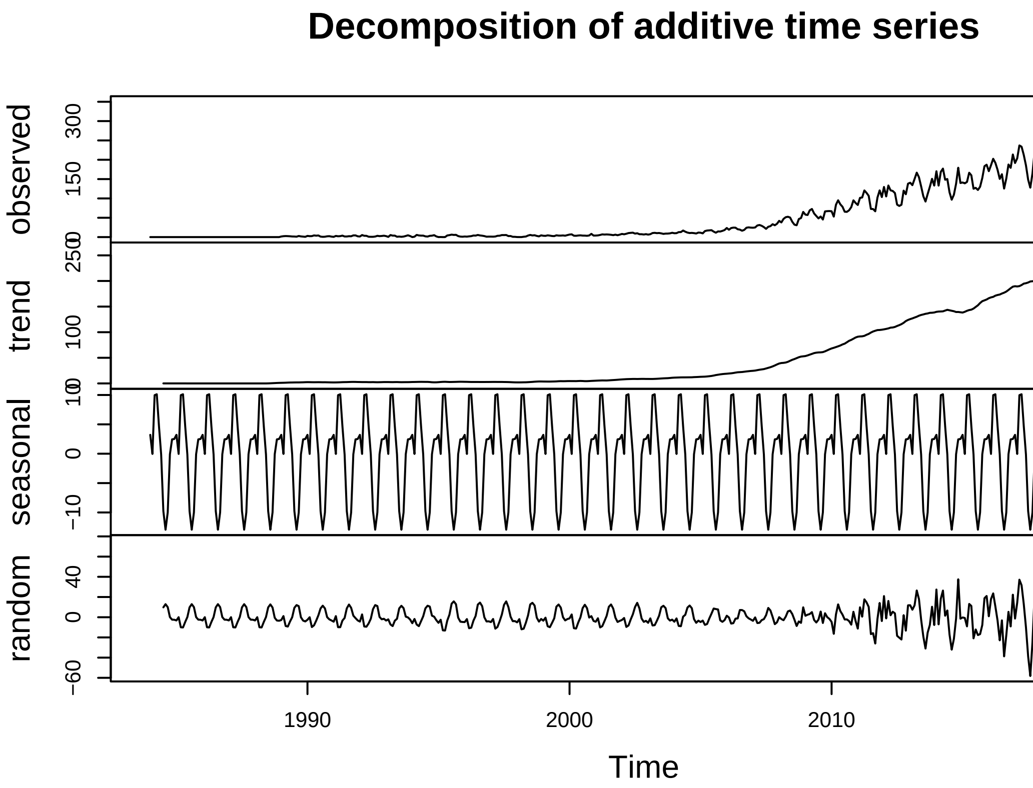


Solar Energy Consumption:

The trend looks to be increasing, but not linearly. We also see two different behaviors: the data is relatively flat until 2012 but then starts to increase at 2012. The random component does not appear to be random, it goes up and down in a very repeated pattern, suggesting some seasonality. Also, it seems like the variance starts to increase around 2005.

```r
wind_decomp <- decompose(energy_ts[,2], type = "additive")
```

```r
# plot trend and random componenet
plot(wind_decomp)
```



Wind Energy Consumption:

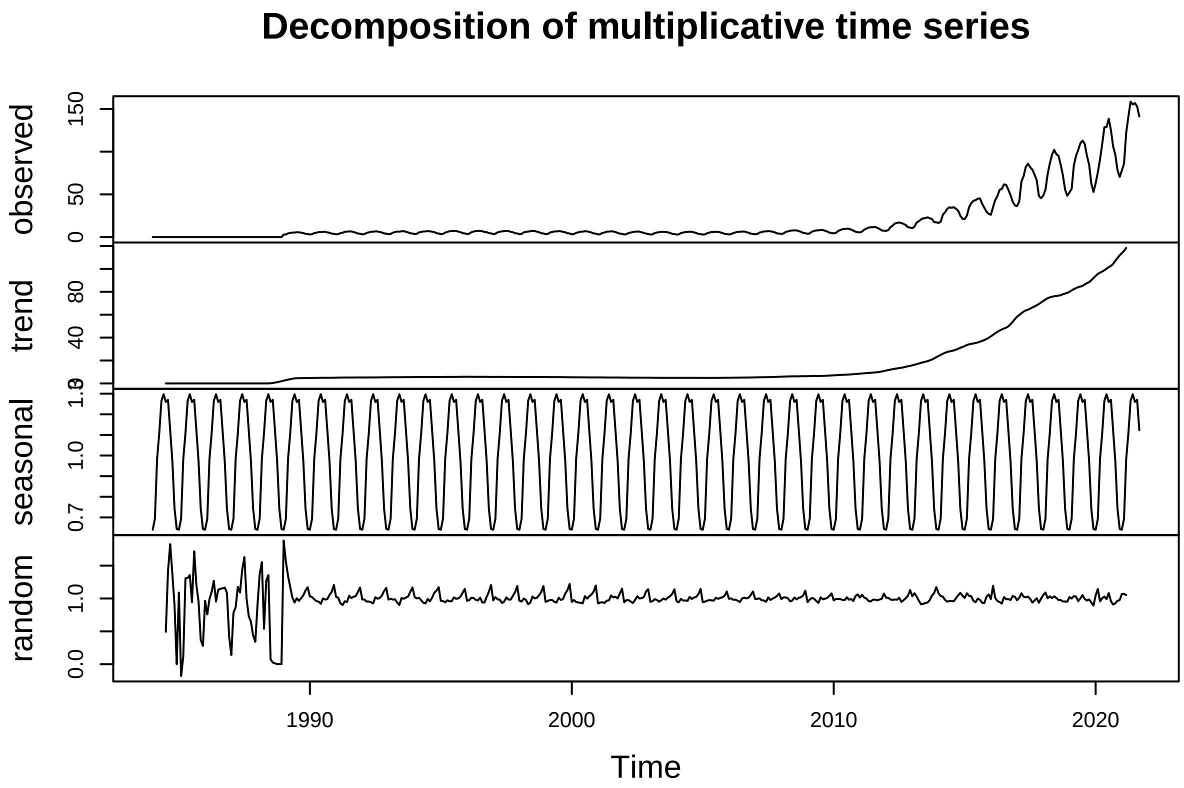
The trend looks to be increasing, but not linearly. Again we see that the data is relatively flat until 2002, but then increases starting in 2002. The random component does not appear to be random, it goes up and down in a very repeated pattern again, suggesting some seasonality. It looks to be more random starting in 2000 however.

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 <- decompose(energy_ts[,1], type = "multiplicative")
```

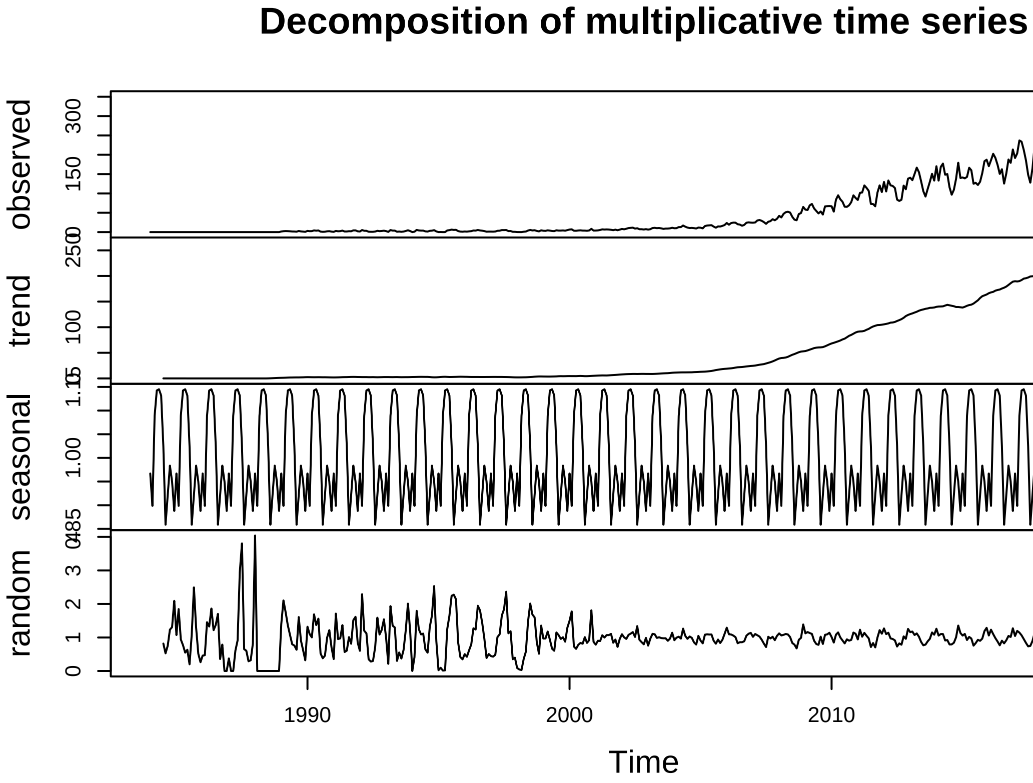
```
# plot trend and random componenet
plot(solar_decomp)
```



The trend still looks to be increasing, but not linearly. Again we see that the data appears to be flat until around 2012 and increasing after. The random component again does not appear to be random, as it still goes up and down in a very repeated pattern, suggesting some seasonality. We do see that from 1984 to around 1990, the random component does look to be much more random.

```
wind_decomp <- decompose(energy_ts[,2], type = "multiplicative")
```

```
# plot trend and random componenet
plot(wind_decomp)
```



Wind Energy Consumption:

The trend looks to be increasing, but not linearly. Again we see that the data appears to be flat until around 2005, and then increasing after. The random component does not appear to be random still. In certain parts, specifically from 1984 to about 2000, it does appear to be more random. However, after 2000, the random componenet looks to have more of a repeated pattern, suggesting seasonality once again.

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, I do not think we need all of the historical data. As I mentioned above, we can see that there are two different behaviors for both trends. For the Solar Energy Data, the data appears relatively flat and close to 0 until 2012, and then it drastically increases after 2012. The Wind Energy Data also appears to be relatively flat and close to 0 until 2002, and then increases after that. Thus, we might not want to use data prior to 2012 for this forecast.

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.

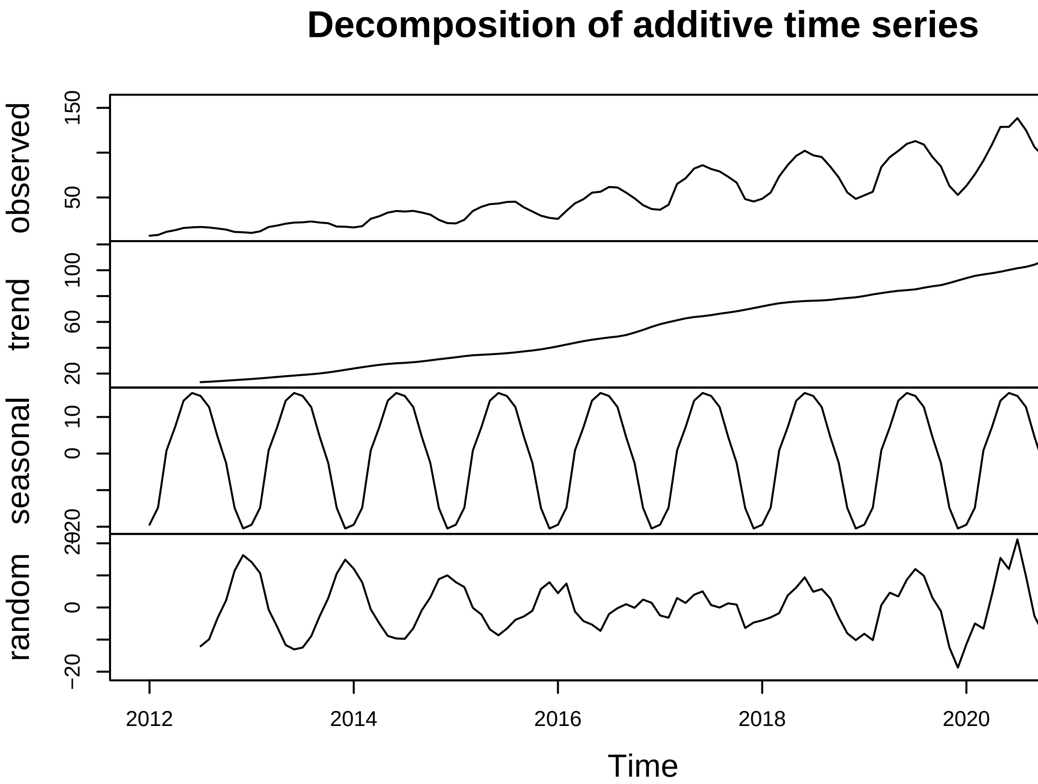
```
# filter for starting in 2012
energy_raw_2012 <- energy_raw %>%
  filter(year(Month) >= 2012)
```

```
# create time series object
energy_ts_2012 <- ts(data = energy_raw_2012 %>% select(-Month), start = c(2012, 1), frequency = 12)
```

Answer:

```
solar_decomp_2012 <- decompose(energy_ts_2012[,1], type = "additive")
```

```
# plot trend and random componenet
plot(solar_decomp_2012)
```

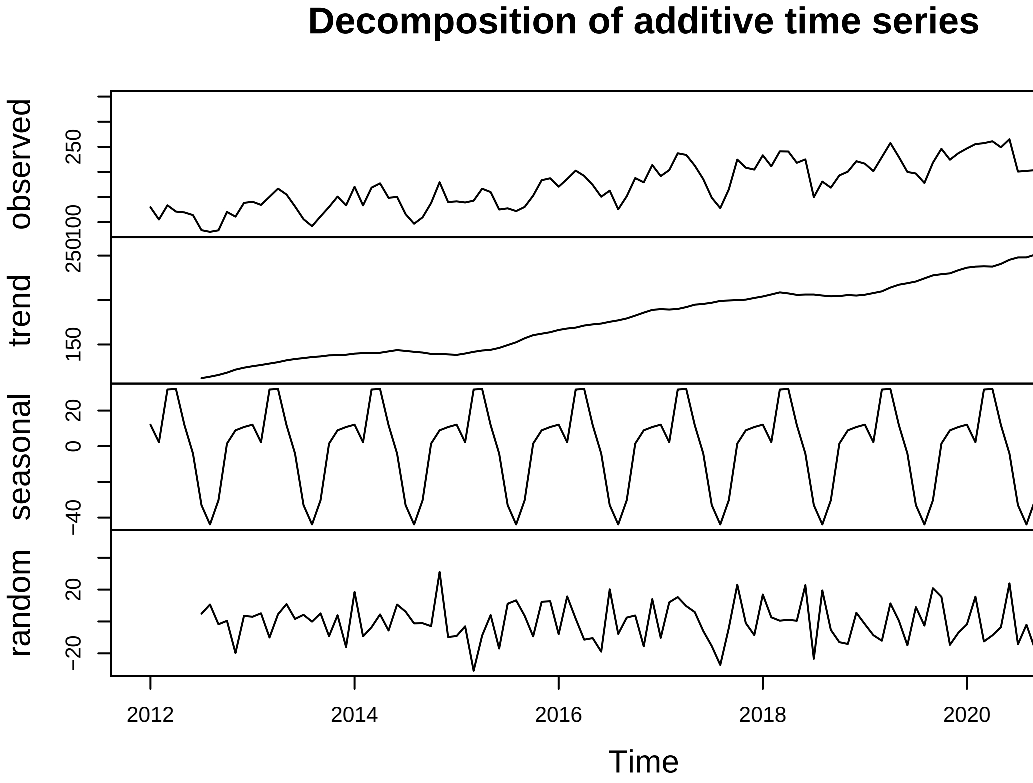


Solar Energy Consumption:

The trend looks to be linearly increasing now. We no longer see the two different behaviors. The random component still does not appear to be super random, it still goes up and down in a somewhat repetitive pattern, suggesting some seasonality.

```
wind_decomp_2012 <- decompose(energy_ts_2012[,2], type = "additive")
```

```
# plot trend and random componenet
plot(wind_decomp_2012)
```



Wind Energy Consumption:

The trend looks to be increasing now as well, and we no longer see the two different behaviors in the trend. The random component appears to be more random now after starting the trend in 2012.