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

Assignment 5 - Due date 02/28/22

Kristen Pulley

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.\

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.

```
#Importing data set - using xlsx package
energy_data<- read_excel("Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx", c
```

```
## New names:
## * `` -> ...1
## * `` -> ...2
## * `` -> ...3
## * `` -> ...4
## * `` -> ...5
## * ...
```

```
#Now let's extract the column names from row 11 only
read_col_names <- read_excel("Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xls
```

```
## New names:
## * `` -> ...1
## * `` -> ...2
## * `` -> ...3
## * `` -> ...4
```

```
## * `` -> ...5
## * ...
```

```
colnames(energy_data) <- read_col_names

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!

```
rawenergy.df <- read_excel("Data/Table_10.1_Renewable_Energy_Production_and_Consumption_by_Source.xlsx")

drop_na(rawenergy.df)
```

```
## # A tibble: 585 x 14
##    Month               `Wood Energy Prod~ `Biofuels Product~ `Total Biomass Ene~
##    <dttm>                           <dbl> <chr>                            <dbl>
##  1 1973-01-01 00:00:00               130. Not Available                     130.
##  2 1973-02-01 00:00:00               117. Not Available                     117.
##  3 1973-03-01 00:00:00               130. Not Available                     130.
##  4 1973-04-01 00:00:00               125. Not Available                     126.
##  5 1973-05-01 00:00:00               130. Not Available                     130.
##  6 1973-06-01 00:00:00               125. Not Available                     126.
##  7 1973-07-01 00:00:00               130. Not Available                     130.
##  8 1973-08-01 00:00:00               130. Not Available                     130.
##  9 1973-09-01 00:00:00               126. Not Available                     126.
## 10 1973-10-01 00:00:00               130. Not Available                     130.
## # ... with 575 more rows, and 10 more variables:
## #   Total Renewable Energy Production <dbl>,
## #   Hydroelectric Power Consumption <dbl>, Geothermal Energy Consumption <dbl>,
## #   Solar Energy Consumption <chr>, Wind Energy Consumption <chr>,
## #   Wood Energy Consumption <dbl>, Waste Energy Consumption <dbl>,
## #   Biofuels Consumption <chr>, Total Biomass Energy Consumption <dbl>,
## #   Total Renewable Energy Consumption <dbl>
```

```
my_date <- paste(rawenergy.df[,1],sep="-")
my_date <- my(my_date)
```

```
## Warning: All formats failed to parse. No formats found.
```

```
energy.df <- cbind(rawenergy.df[,c(1,8,9)])
names(energy.df)<-c('Time','Solar','Wind')

energy.df$Solar<-as.numeric(energy.df$Solar)
```

```
## Warning: NAs introduced by coercion
```

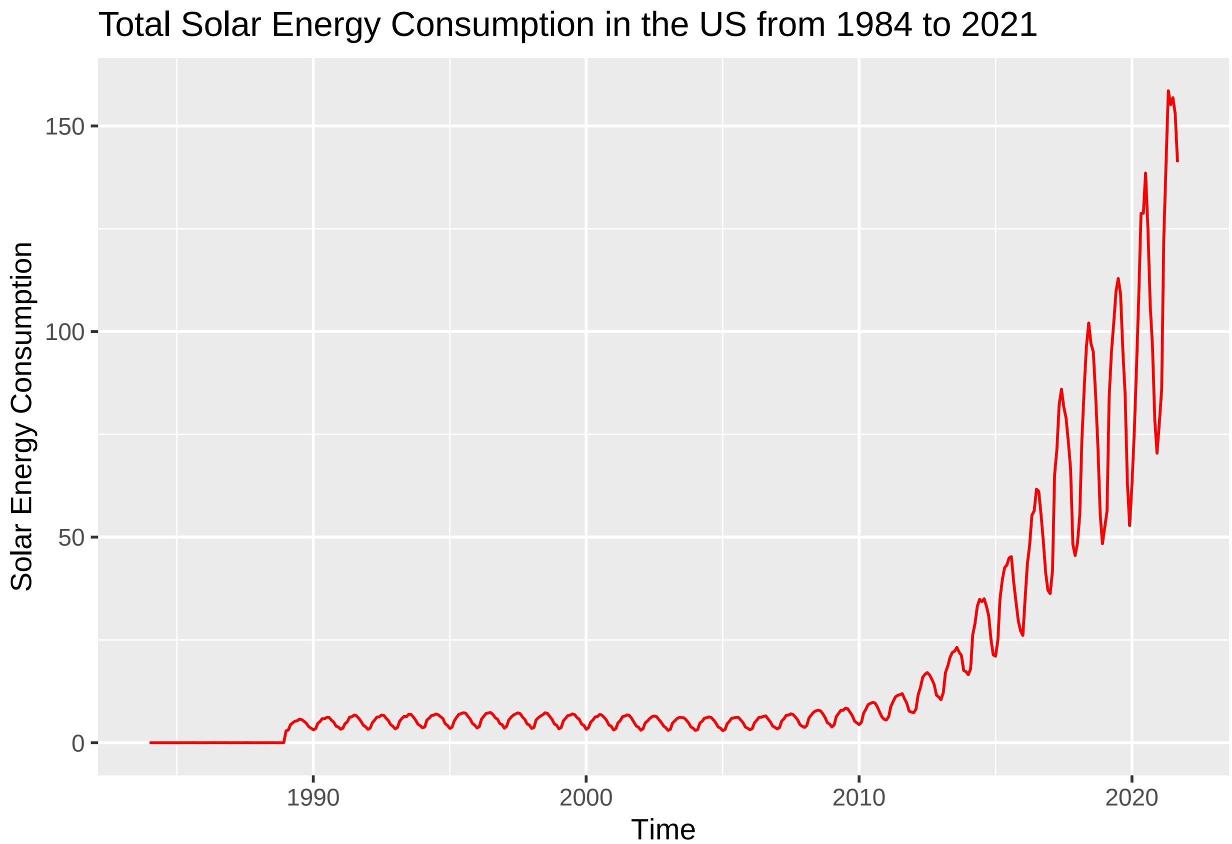
```
energy.df$Wind<-as.numeric(energy.df$Wind)
```

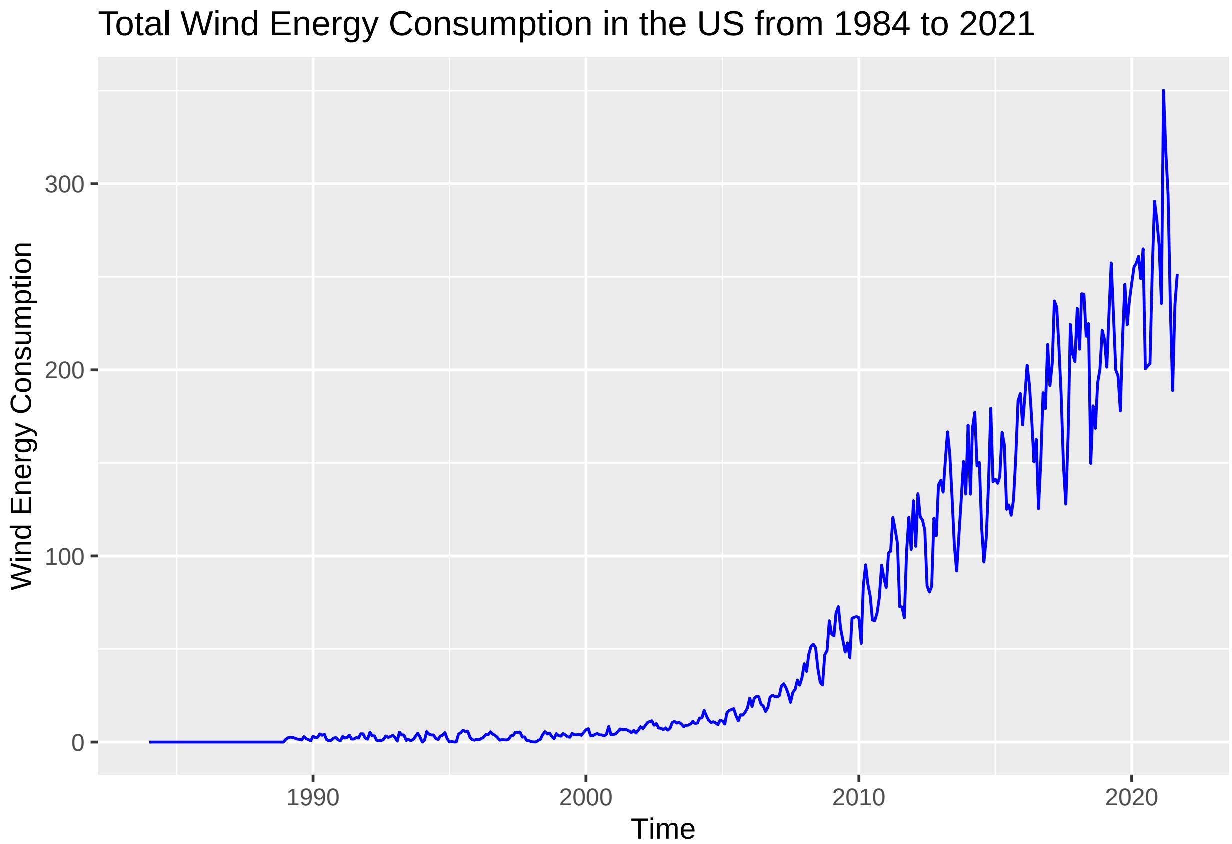
```
## Warning: NAs introduced by coercion
```

```
energy.df<-drop_na(energy.df)
```

Q2

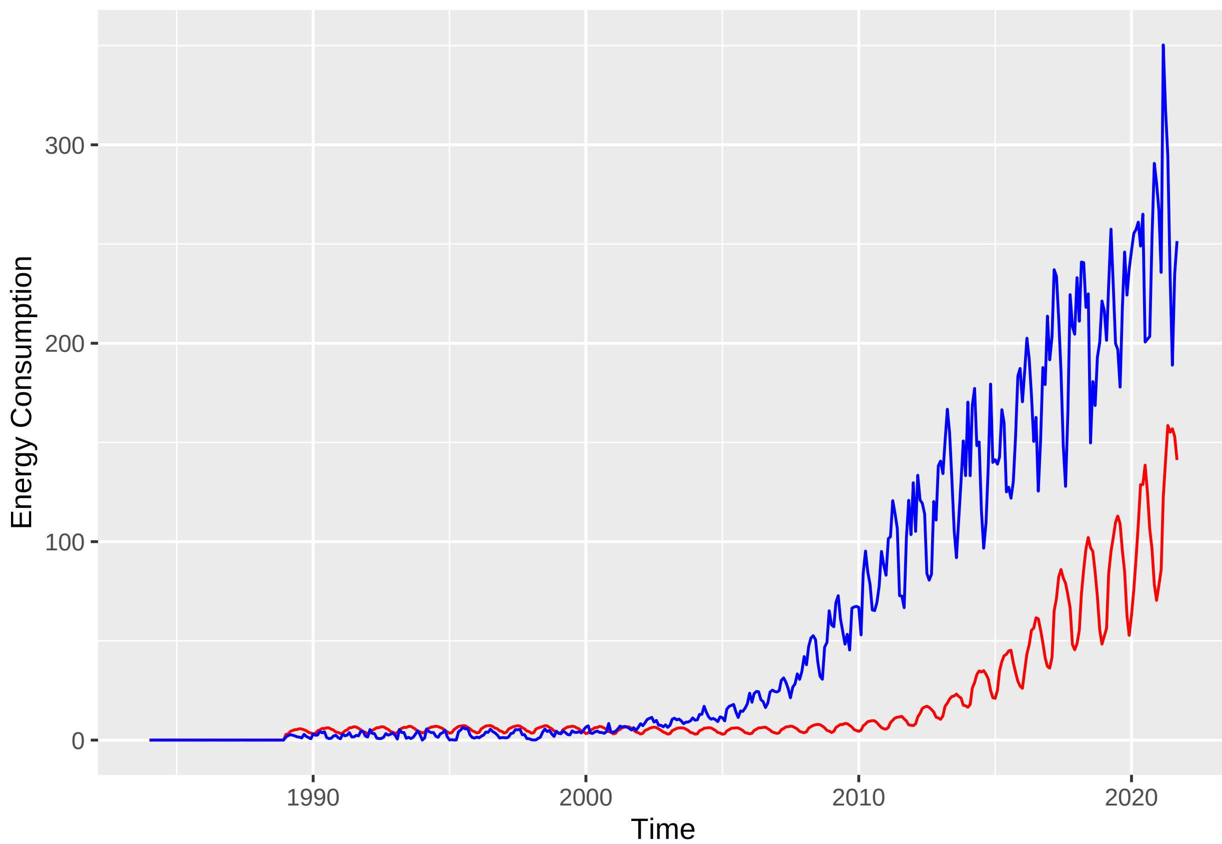
Plot the Solar and Wind energy consumption over time using ggplot. Plot each series on a separate graph. No need to add legend. Add informative names to the y axis using `ylab()`. Explore the function scale_x_date() on ggplot and see if you can change the x axis to improve your plot. Hint: use *scale_x_date(date_breaks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





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.



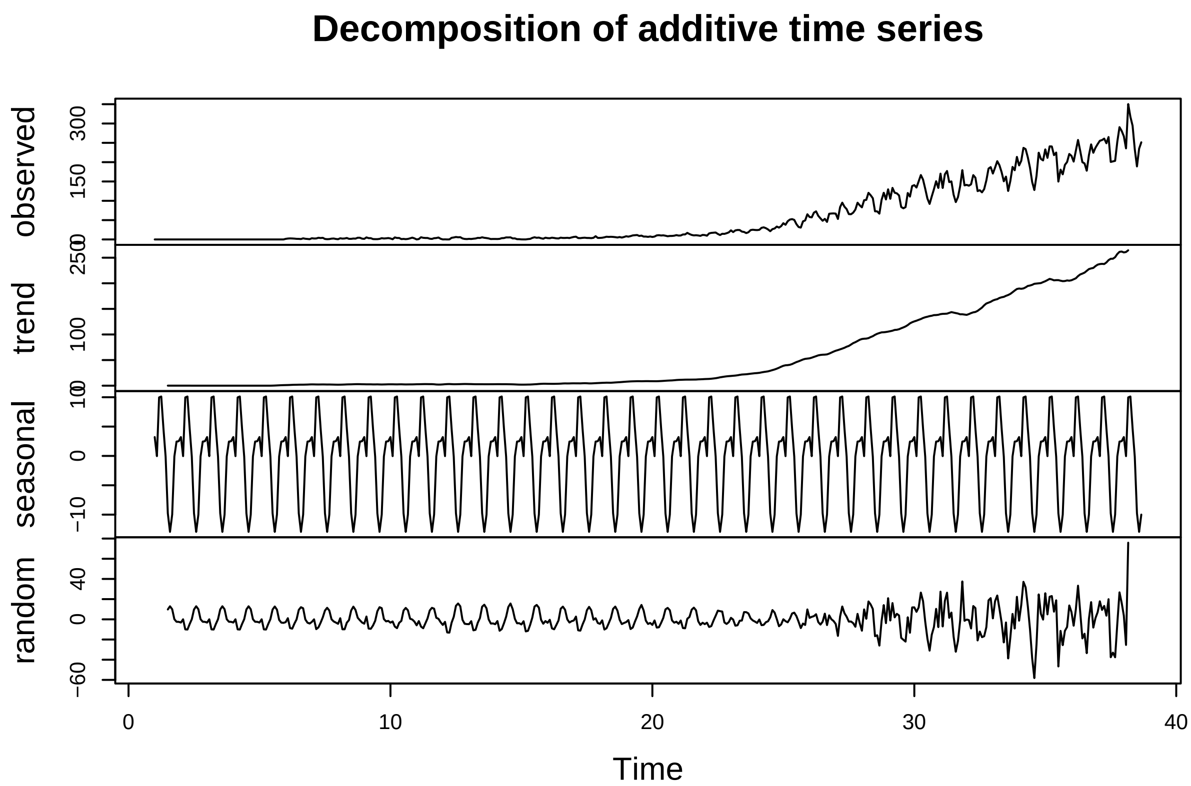
Q3

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? > Answer: The trend for both the wind and solar data have an obvious positive upwards trend. Both random components appear to have seasonality in it as seen by the continuous and increasing frequency of the line.

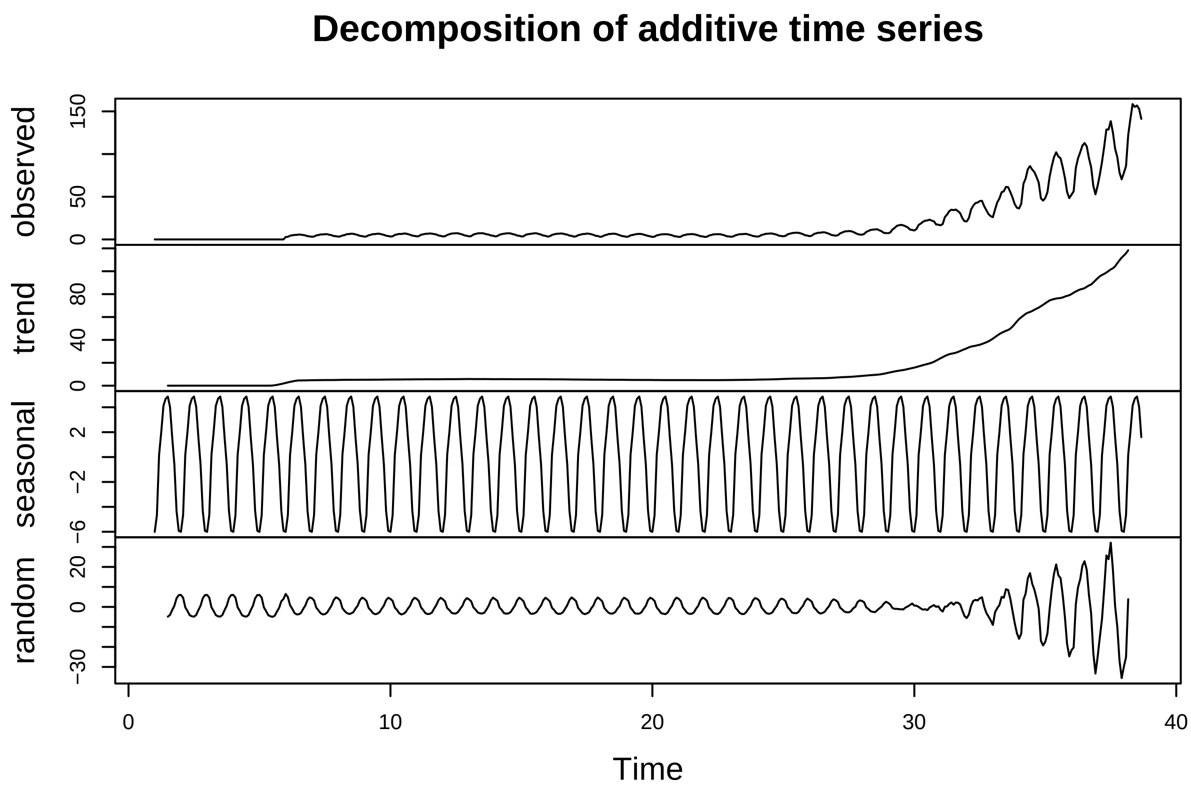
```
solar_ts<-ts(energy.df[,2], frequency = 12)
wind_ts<-ts(energy.df[,3], frequency = 12)

decomp_solar<-decompose(solar_ts, type = "additive")
decomp_wind<-decompose(wind_ts, type = "additive")

  plot(decomp_wind)
```




```
plot(decomp_solar)
```

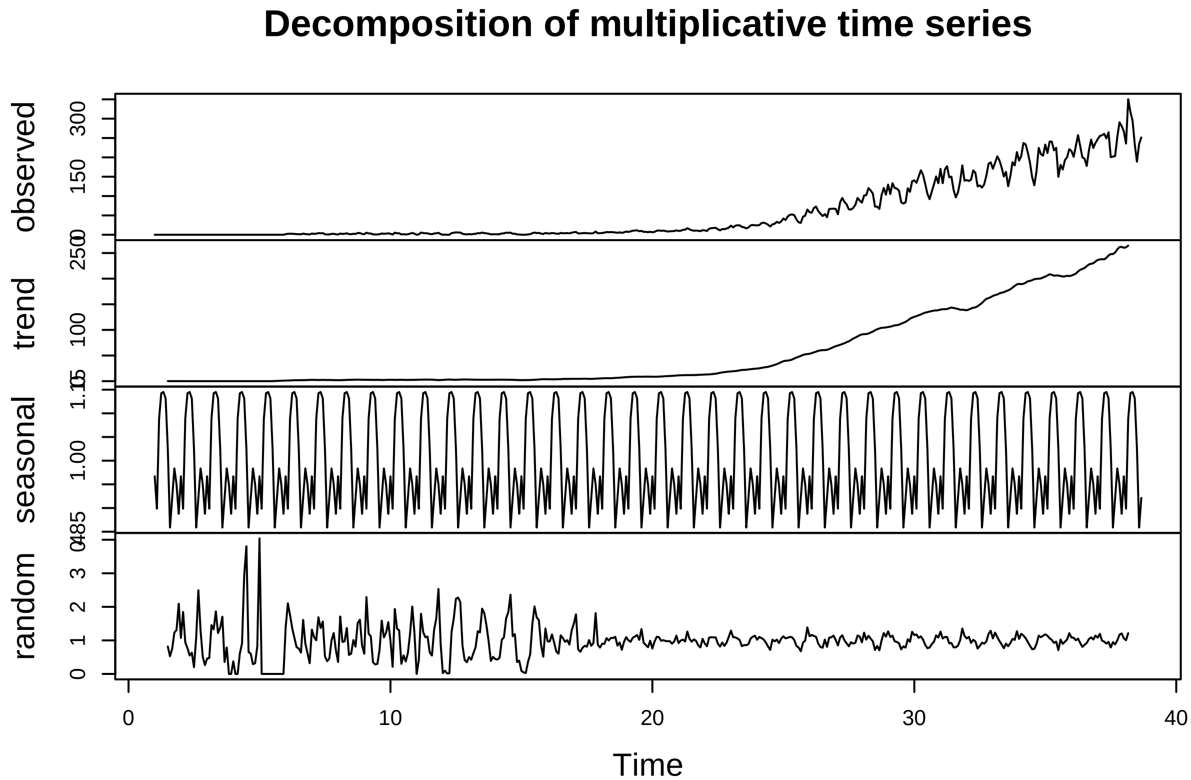


Q4

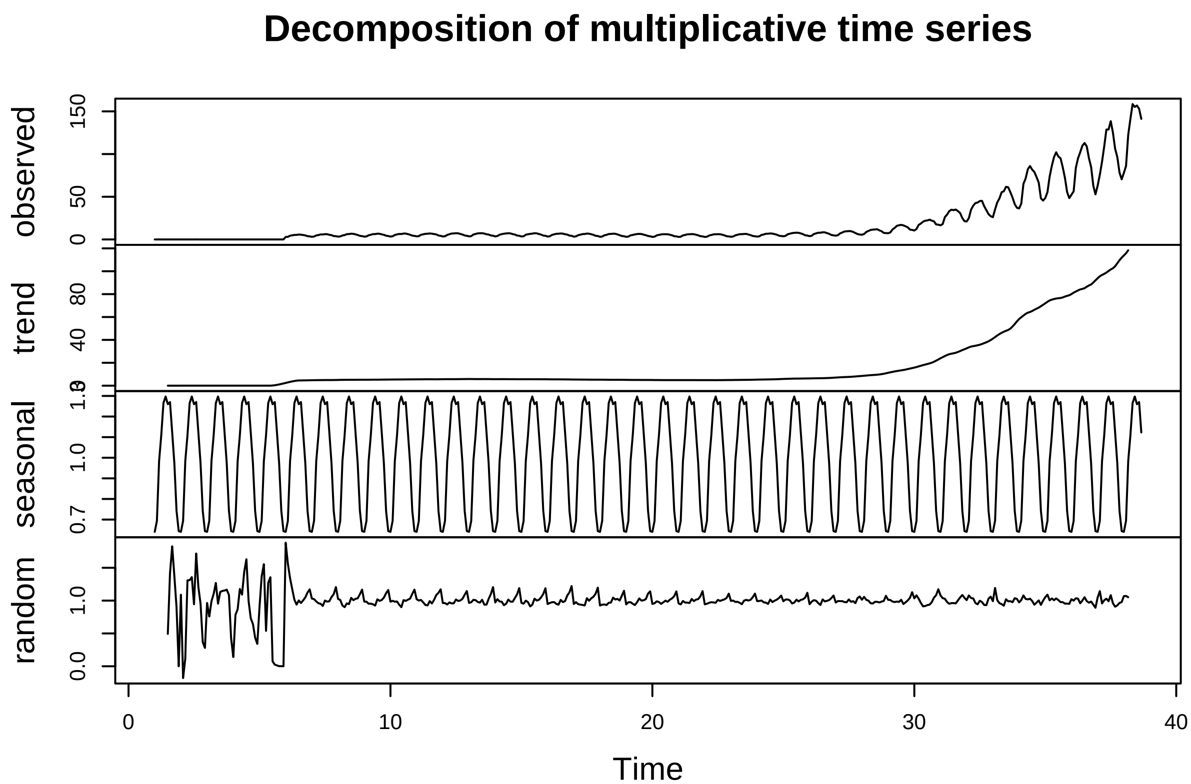
Use the decompose function again but now change the type of the seasonal component from additive to multiplicative. What happened to the random component this time? > Answer: It looks like the random component is reveresed for both the solar and wind data. However, the wind data shows more of a continuous trend than the solar data.

```
decomp2_solar<-decompose(solar_ts, type = "multiplicative")
decomp2_wind<-decompose(wind_ts, type = "multiplicative")

  plot(decomp2_wind)
```




```
plot(decomp2_solar)
```



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.

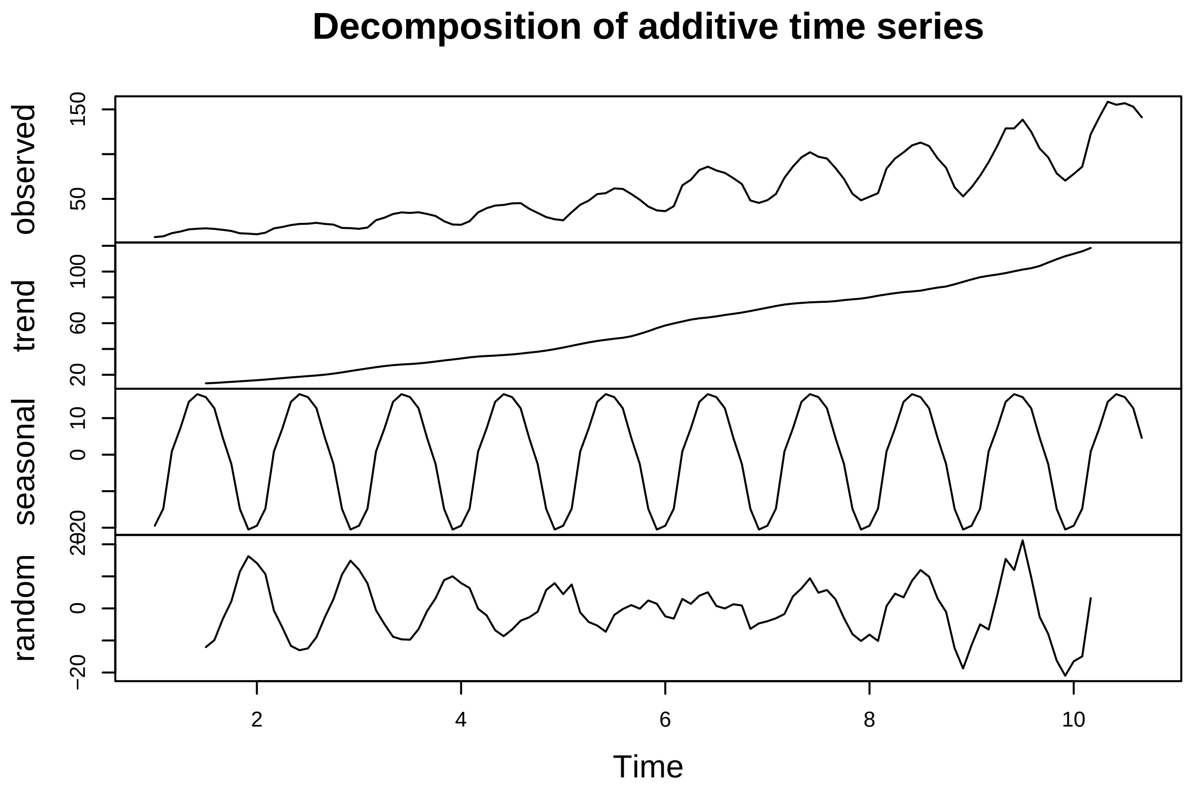
> Answer: I think data from the 90's would be raltively unimportant because of the slow development of these renewable energies. Inlcuding this data would alter the trend line of a regression analysis. I would consider them level shift outliers, because we know that trend will never go back to that. Solar/wind energy consumption will only continue to rise. Inlcuding it would probably lead to an underestimating forecast of future consumption.

Q6

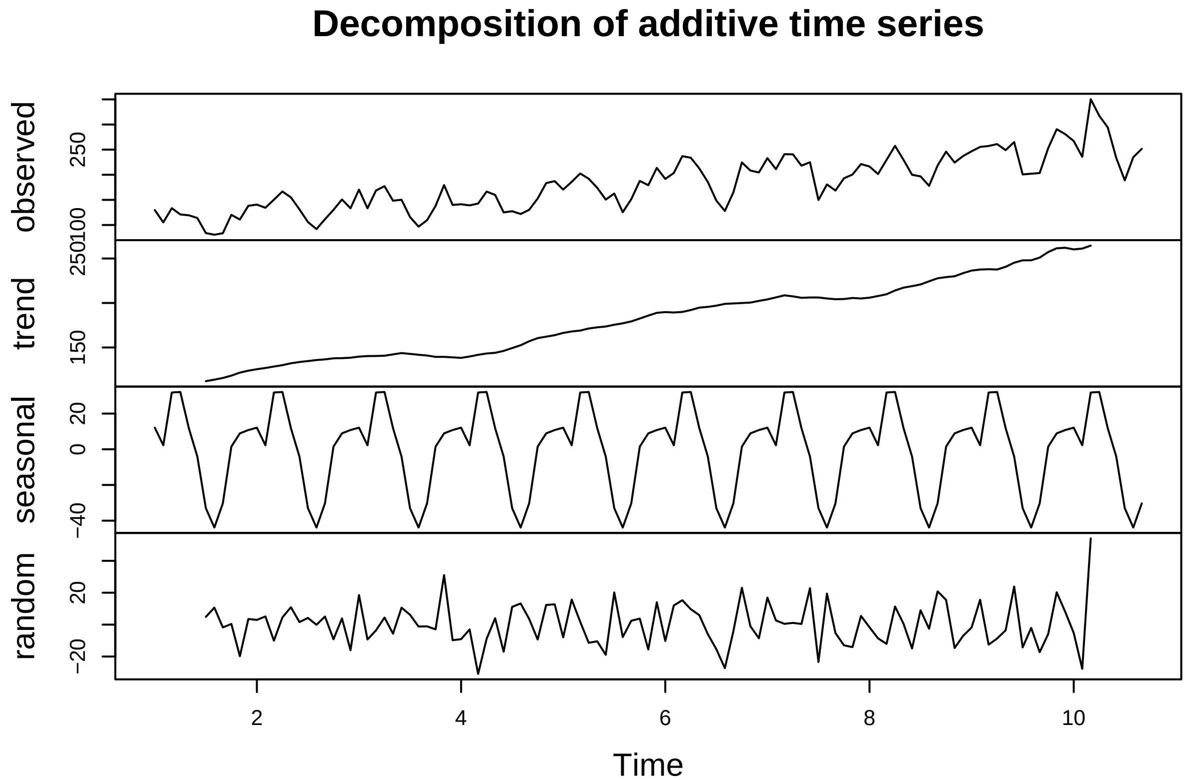
Create a new time series object where historical data starts on January 2012. Hint: use `filter()` function so that you don't need to point to row numbers, .i.e, `filter(xxxx, year(Date) >= 2012 )`. Apply the decompose function `type=additive` to this new time series. Comment the results. Does the random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

```
newenergy.df<-filter(energy.df, year(Time)>=2012)
newsun.ts<-ts(newenergy.df$Solar, frequency = 12)
newwind.ts<-ts(newenergy.df$Wind, frequency = 12)
decompsun_new<-decompose(newsun.ts, type = "additive")
```

```r
decompwind_new<-decompose(newwind.ts, type = "additive")
plot(decompsun_new)
```




```r
plot(decompwind_new)
```



Answer: The random component looks a lot more random compared to the graphs above. There is less of a seasonality trend evident in that component. Because I identified the outliers prior to 2012 as a level shift, filtering them out has probably done more to avpid distorting the seasonal component, so that the data may be forecasted accurately.