

# ENV 790.30 - Time Series Analysis for Energy Data | Spring 2024

Assignment 5 - Due date 02/19/24

Julia

## 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\_Sp23.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)
```

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

```
library(readxl)  
library(stats)  
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)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --  
## v dplyr   1.1.3      v stringr 1.5.0  
## v forcats 1.0.0      v tibble  3.2.1  
## v purrr   1.0.2      v tidyr   1.3.0
```

```
## v readr 2.1.4

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag() masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(cowplot)

##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
## stamp
```

## 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 December 2023 Monthly Energy Review.

```
#Importing data set - using xlsx package
energy_data <- read_excel("~/Julia_Kagiliery_TSA_Sp24/Data/Table_10.1_Renewable_Energy_Production_and_Consumption.xlsx",
  skip = 9)
energy_data <- energy_data[-1,]
```

### 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(c("Solar Energy Consumption", "Wind Energy Consumption", "Month"))

energy_data <- energy_data |>
  filter(
    `Solar Energy Consumption` != "Not Available",
    `Wind Energy Consumption` != "Not Available")

energy_data <- na.omit(energy_data)

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

Chat GPT prompt for `ymd()`: how to use `my()` function for date conversions

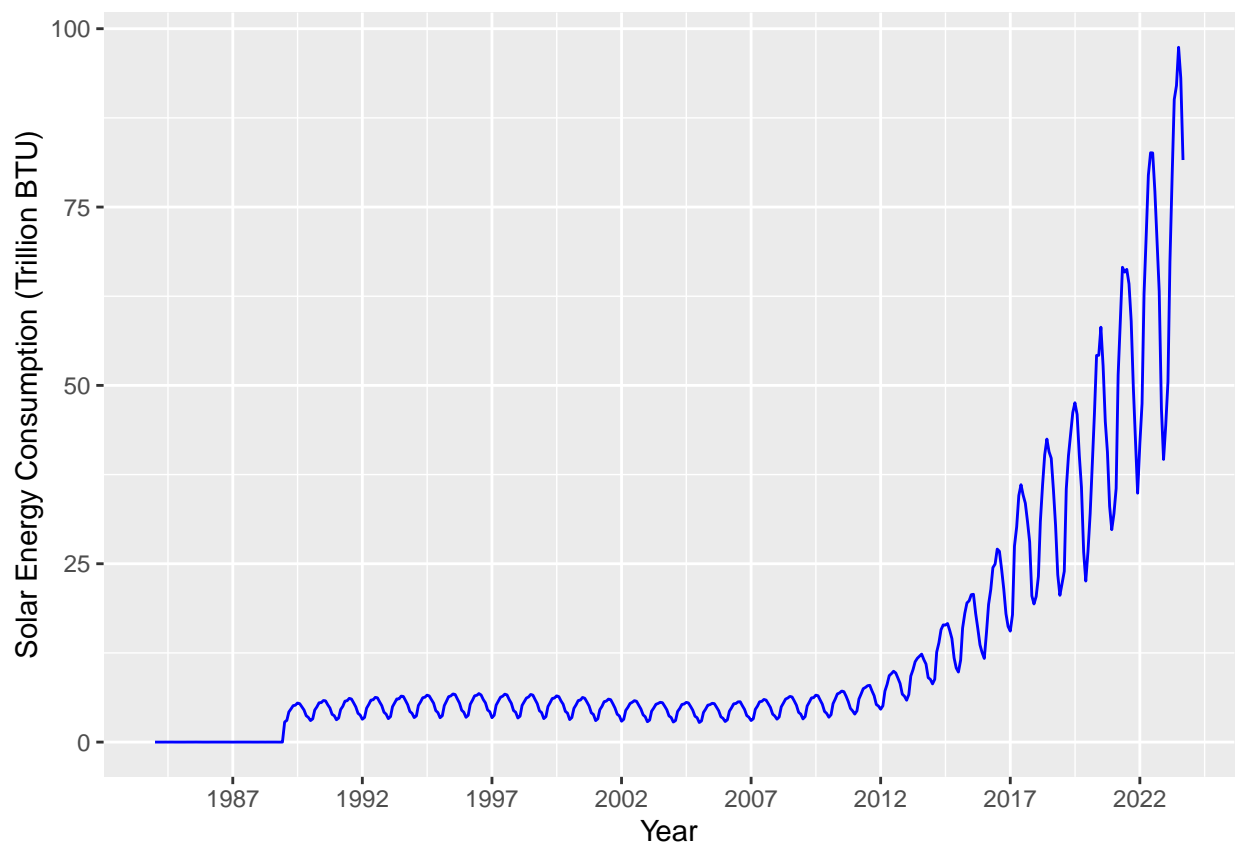
```
nobs=nrow(energy_data)
nvar=ncol(energy_data)
```

```
energyTS <- ts(energy_data)
```

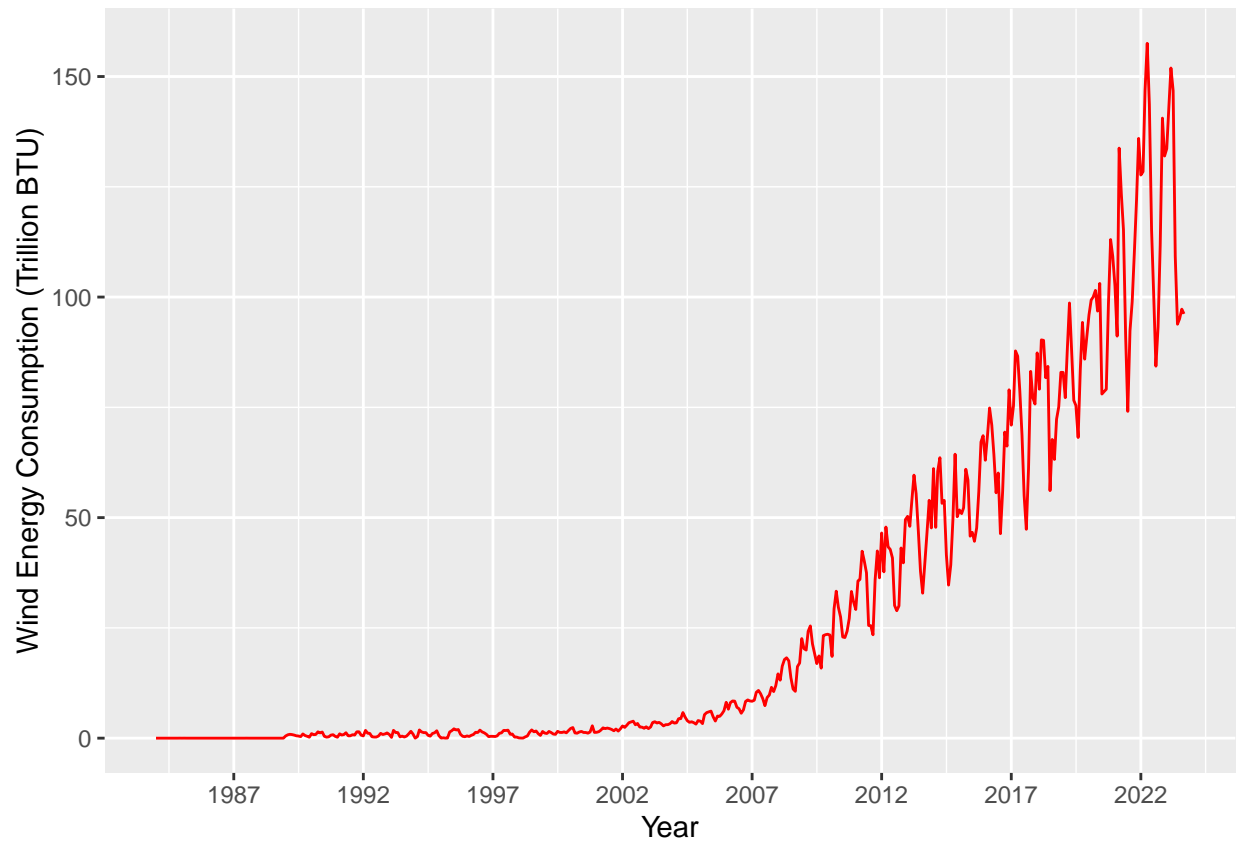
## Q2

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")`

```
energy_data |>
ggplot(aes(x=Month, y= `Solar Energy Consumption`)) +
  geom_line(color="blue") +
  scale_x_date(date_breaks = "5 years", date_labels = "%Y") +
  ylab("Solar Energy Consumption (Trillion BTU)") +
  xlab("Year")
```



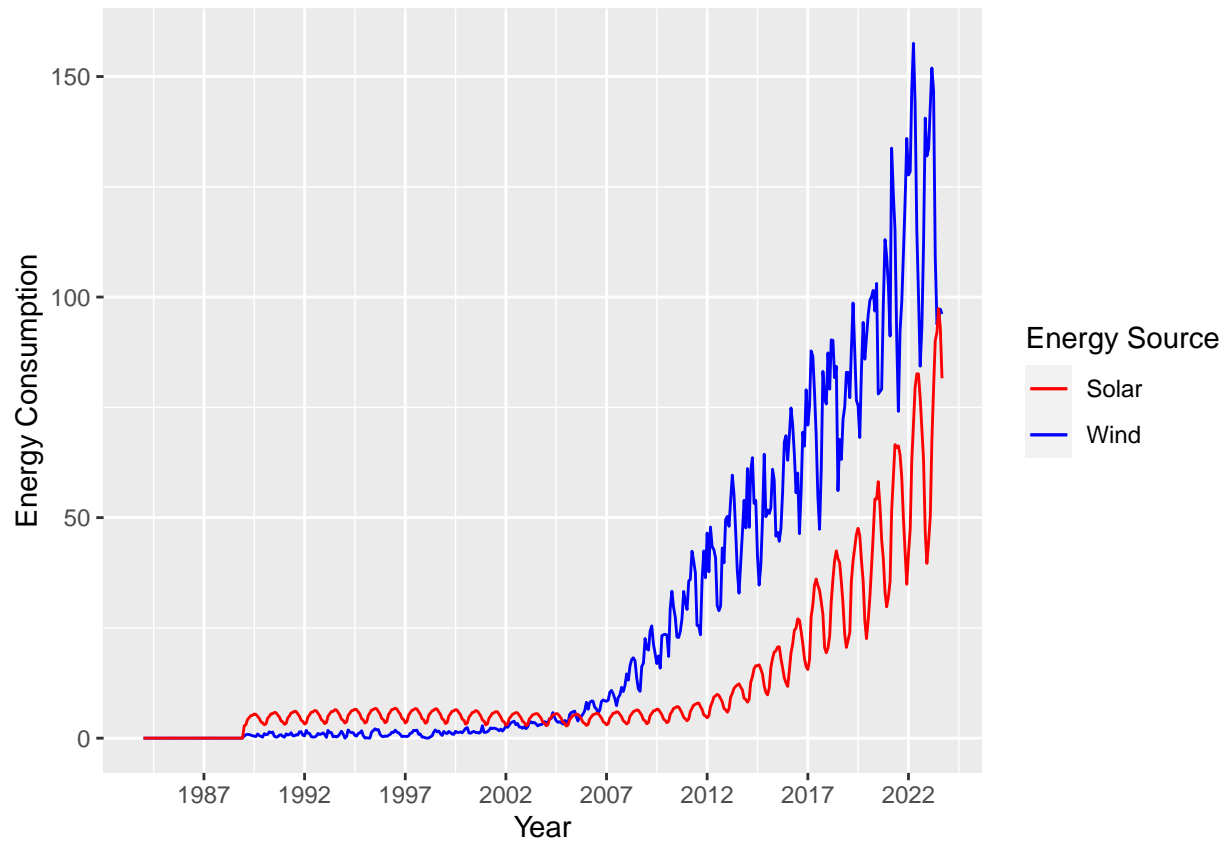
```
energy_data |>
ggplot(aes(x=Month, y= `Wind Energy Consumption`)) +
  geom_line(color="red") +
  scale_x_date(date_breaks = "5 years", date_labels = "%Y") +
  ylab("Wind Energy Consumption (Trillion BTU)") +
  xlab("Year")
```



### 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.

```
energy_data |>
  ggplot(aes(x = Month, color = factor("Wind"), y = `Wind Energy Consumption`)) +
  geom_line() +
  geom_line(aes(x = Month, color = factor("Solar"), y = `Solar Energy Consumption`)) +
  scale_x_date(date_breaks = "5 years", date_labels = "%Y") +
  ylab("Energy Consumption") +
  xlab("Year") +
  scale_color_manual(name = "Energy Source", values = c("Wind" = "blue", "Solar" = "red"))
```



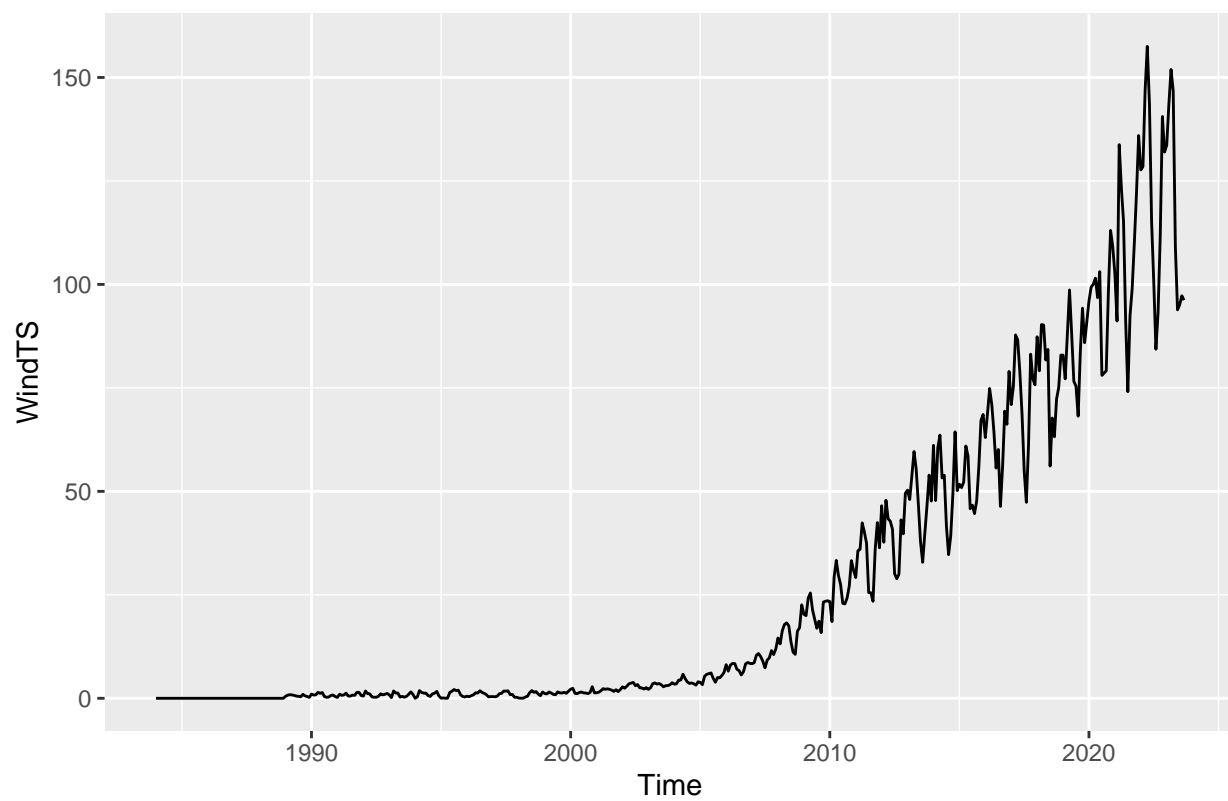
Code inspo: Chat GPT. Prompt: why isn't "scale\_color\_manual(values = colors)" adding a legend

## Decomposing the time series

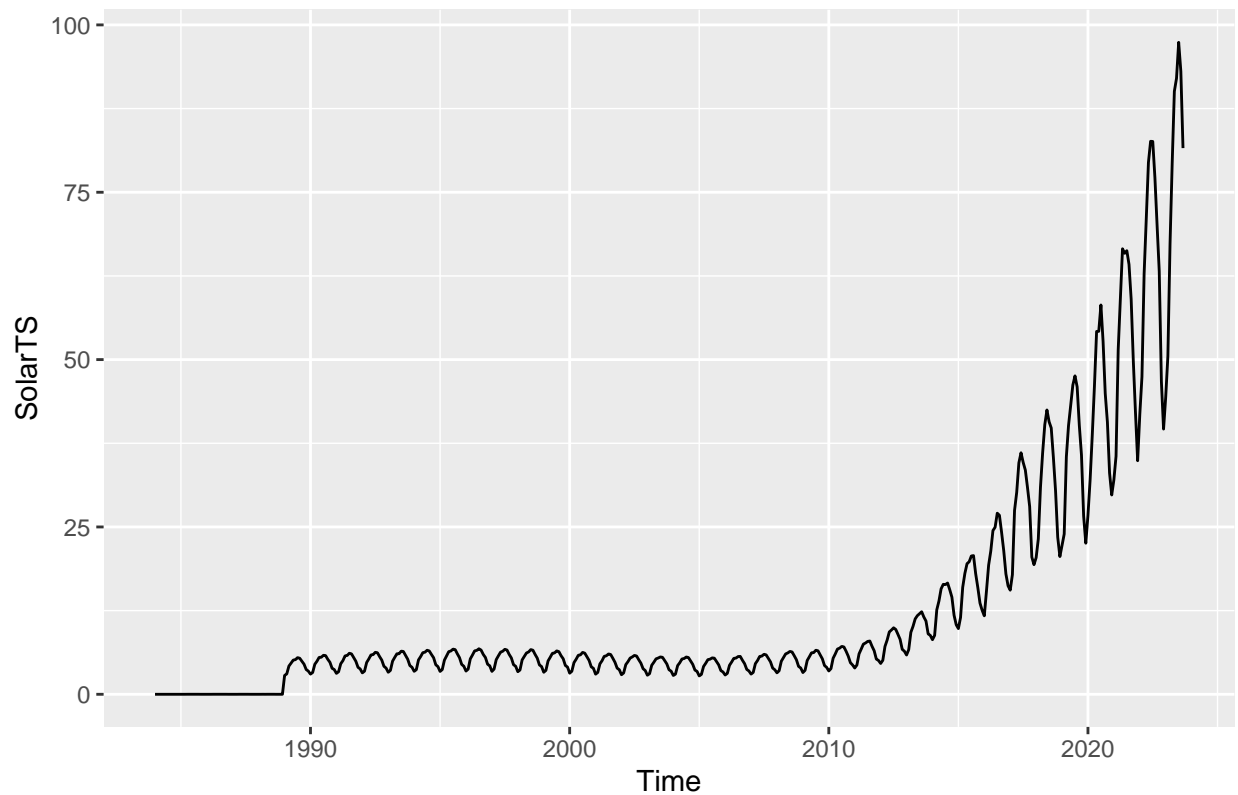
The stats package has a function called `decompose()`. This function only take time series object. As the name says the decompose function will decompose your time series into three components: trend, seasonal and random. This is similar to what we did in the previous script, but in a more automated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

```
WindTS <- ts(energy_data$`Wind Energy Consumption`, start = c(1984,1), frequency = 12)
SolarTS <- ts(energy_data$`Solar Energy Consumption`, start = c(1984,1), frequency = 12)

autoplot(WindTS)
```



```
autoplot(SolarTS)
```



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.

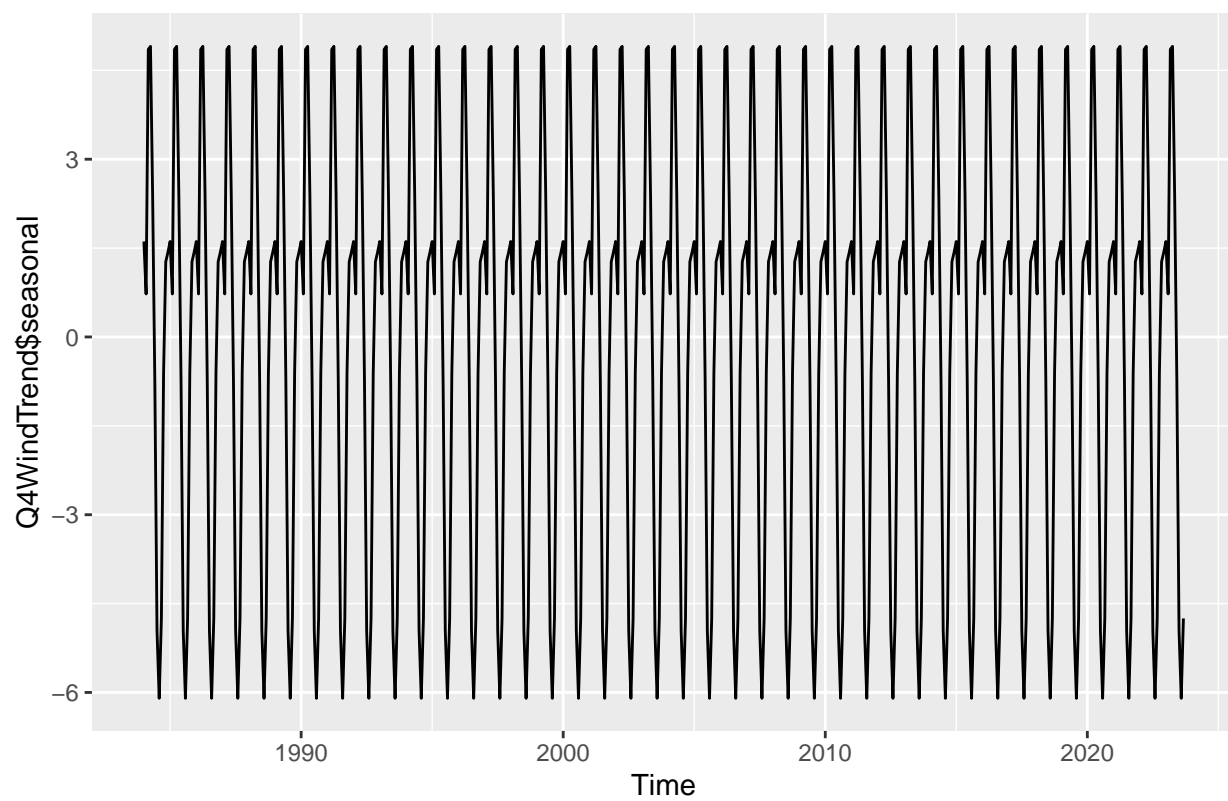
#### Q4

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?

Answer: The trend component looks about how I would expect it: increasing. However, the random component does not look random at all; there certainly seems to be some seasonality left. The random plots almost seem to imply that data before 2012 is one model and data after 2012 is a totally different one.

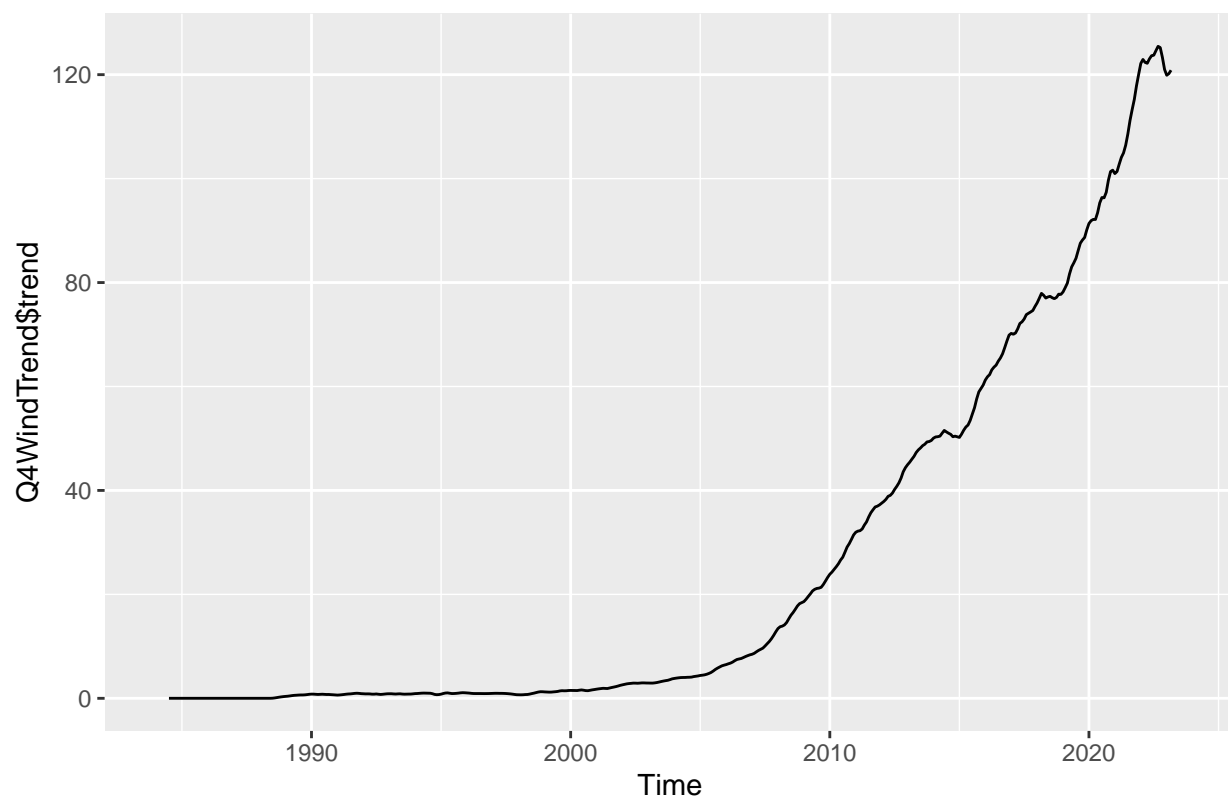
```
Q4WindTrend <- decompose(WindTS, type = "additive")
```

```
autoplot(Q4WindTrend$seasonal)
```

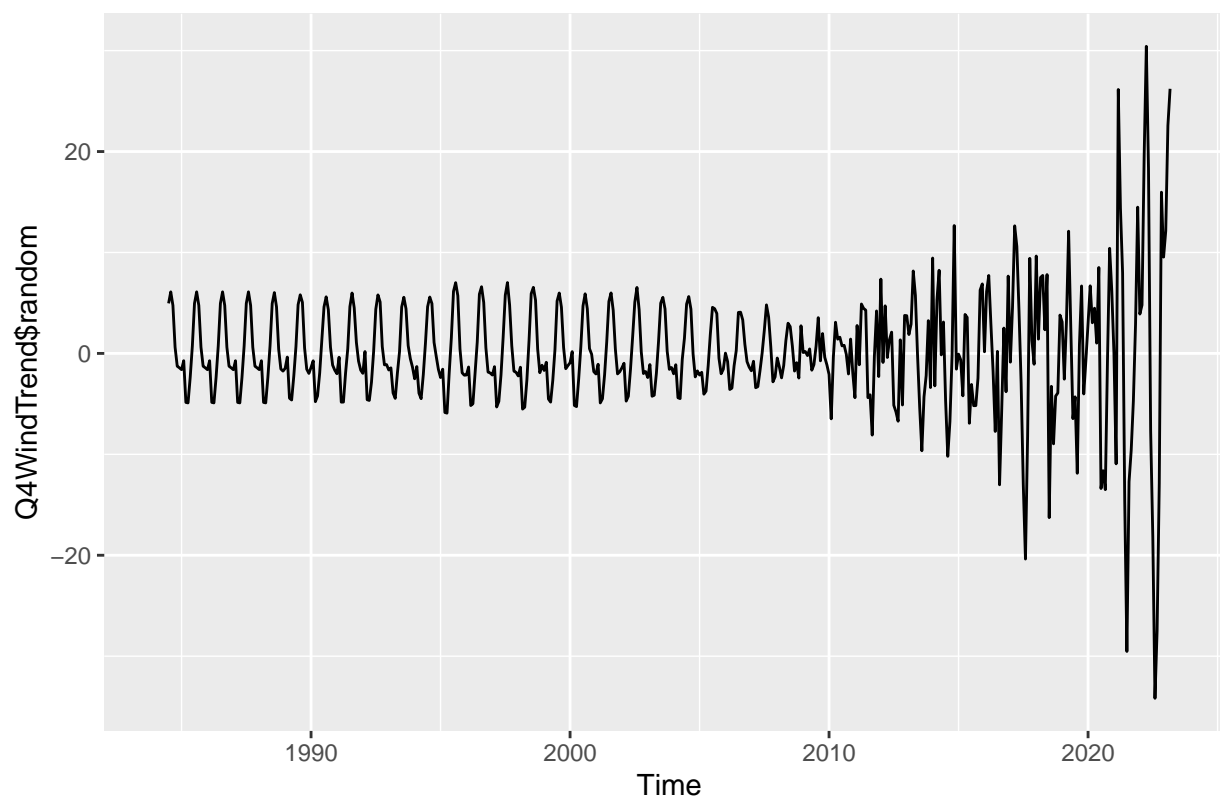


```
autoplot(Q4WindTrend$trend)
```





```
autoplot(Q4WindTrend$random)
```



```
Q4SolarTrend <- print(decompose(SolarTS, type = "additive"))
```

```
## $x
##      Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct
## 1984  0.000  0.000  0.001  0.001  0.002  0.003  0.001  0.003  0.003  0.002
## 1985  0.002  0.000  0.000  0.004  0.005  0.005  0.004  0.007  0.005  0.003
## 1986  0.002  0.002  0.003  0.005  0.006  0.005  0.006  0.006  0.005  0.004
## 1987  0.000  0.002  0.003  0.004  0.004  0.006  0.006  0.004  0.003  0.002
## 1988  0.002  0.003  0.004  0.002  0.004  0.005  0.004  0.004  0.003  0.000
## 1989  2.821  3.041  4.220  4.660  5.109  5.175  5.476  5.389  4.941  4.501
## 1990  3.015  3.256  4.489  4.946  5.515  5.534  5.832  5.781  5.228  4.782
## 1991  3.125  3.369  4.561  5.026  5.739  5.803  6.133  6.035  5.487  4.935
## 1992  3.196  3.451  4.752  5.284  5.883  5.931  6.274  6.205  5.628  5.115
## 1993  3.288  3.570  4.954  5.515  6.053  6.058  6.445  6.401  5.818  5.243
## 1994  3.432  3.680  5.137  5.640  6.216  6.292  6.584  6.471  5.949  5.458
## 1995  3.415  3.692  5.105  5.735  6.370  6.471  6.750  6.674  6.047  5.471
## 1996  3.472  3.760  5.293  5.865  6.487  6.562  6.804  6.602  5.956  5.494
## 1997  3.422  3.762  5.176  5.748  6.292  6.433  6.721  6.612  5.958  5.437
## 1998  3.375  3.629  5.138  5.664  6.173  6.327  6.679  6.584  5.956  5.395
## 1999  3.294  3.570  4.984  5.536  6.121  6.184  6.482  6.377  5.775  5.311
## 2000  3.155  3.440  4.733  5.239  5.832  5.895  6.261  6.144  5.608  5.056
## 2001  3.008  3.261  4.542  5.020  5.672  5.755  6.029  5.947  5.380  4.773
## 2002  2.923  3.189  4.435  4.877  5.408  5.585  5.812  5.719  5.162  4.647
## 2003  2.848  3.083  4.334  4.795  5.298  5.421  5.579  5.522  5.034  4.535
## 2004  2.809  3.019  4.289  4.722  5.278  5.343  5.575  5.488  4.986  4.470
## 2005  2.742  2.968  4.158  4.638  5.178  5.247  5.438  5.396  4.894  4.399
```

##	2006	2.881	3.124	4.327	4.821	5.368	5.410	5.640	5.657	5.085	4.577
##	2007	3.016	3.269	4.579	5.051	5.658	5.707	5.981	5.886	5.367	4.848
##	2008	3.206	3.518	4.933	5.478	6.029	6.181	6.408	6.326	5.758	5.166
##	2009	3.247	3.578	5.053	5.614	6.200	6.229	6.582	6.501	5.892	5.307
##	2010	3.485	3.833	5.386	6.033	6.760	6.895	7.158	7.072	6.433	5.693
##	2011	3.924	4.374	6.043	6.760	7.496	7.667	7.903	7.958	7.178	6.507
##	2012	4.607	5.077	7.148	8.096	9.316	9.605	9.934	9.685	8.960	8.214
##	2013	5.869	6.663	9.260	10.151	11.264	11.745	12.038	12.336	11.551	10.946
##	2014	8.157	8.799	12.624	13.934	15.758	16.428	16.395	16.624	15.631	14.507
##	2015	9.815	11.480	15.989	18.058	19.510	19.804	20.660	20.720	18.026	15.938
##	2016	11.728	15.428	19.297	21.401	24.459	24.955	27.056	26.741	24.199	21.438
##	2017	15.555	17.857	27.472	30.175	34.567	36.083	34.635	33.492	30.881	28.042
##	2018	20.417	23.213	30.918	36.049	40.277	42.476	40.715	39.785	35.355	30.386
##	2019	22.249	23.942	35.490	40.146	43.146	46.198	47.572	45.914	40.157	35.724
##	2020	26.741	32.049	38.731	46.045	54.208	54.219	58.159	52.712	44.933	40.674
##	2021	32.034	35.565	51.477	59.068	66.559	65.882	66.269	64.229	59.026	49.778
##	2022	41.808	47.446	62.806	71.072	79.459	82.611	82.584	77.169	70.105	63.190
##	2023	44.432	50.523	67.312	79.380	90.079	92.024	97.397	93.068	81.595	
##		Nov	Dec								
##	1984	0.001	0.000								
##	1985	0.001	0.001								
##	1986	0.003	0.001								
##	1987	0.001	0.001								
##	1988	0.000	0.000								
##	1989	3.697	3.465								
##	1990	3.884	3.700								
##	1991	4.034	3.792								
##	1992	4.187	3.914								
##	1993	4.297	4.039								
##	1994	4.415	4.136								
##	1995	4.473	4.192								
##	1996	4.495	4.262								
##	1997	4.424	4.187								
##	1998	4.389	4.147								
##	1999	4.307	4.068								
##	2000	4.103	3.882								
##	2001	3.905	3.675								
##	2002	3.821	3.551								
##	2003	3.669	3.452								
##	2004	3.625	3.418								
##	2005	3.547	3.335								
##	2006	3.718	3.487								
##	2007	3.922	3.662								
##	2008	4.167	3.926								
##	2009	4.300	4.023								
##	2010	4.721	4.384								
##	2011	5.259	5.056								
##	2012	6.715	6.439								
##	2013	9.028	8.800								
##	2014	11.805	10.387								
##	2015	13.628	12.547								
##	2016	17.985	16.202								
##	2017	20.501	19.362								
##	2018	23.468	20.576								

```

## 2019 26.634 22.573
## 2020 33.068 29.778
## 2021 42.082 34.895
## 2022 46.708 39.623
## 2023
##
## $seasonal
##           Jan           Feb           Mar           Apr           May           Jun
## 1984 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1985 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1986 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1987 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1988 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1989 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1990 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1991 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1992 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1993 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1994 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1995 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1996 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1997 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1998 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 1999 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2000 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2001 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2002 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2003 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2004 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2005 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2006 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2007 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2008 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2009 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2010 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2011 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2012 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2013 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2014 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2015 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2016 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2017 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2018 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2019 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2020 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2021 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2022 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
## 2023 -4.00420660 -3.15722904 0.03232545 1.40253766 2.80309139 3.06251025
##           Jul           Aug           Sep           Oct           Nov           Dec
## 1984  3.21594084  2.64254447  1.13678165 -0.36339998 -2.82600681 -3.94488929
## 1985  3.21594084  2.64254447  1.13678165 -0.36339998 -2.82600681 -3.94488929
## 1986  3.21594084  2.64254447  1.13678165 -0.36339998 -2.82600681 -3.94488929
## 1987  3.21594084  2.64254447  1.13678165 -0.36339998 -2.82600681 -3.94488929
## 1988  3.21594084  2.64254447  1.13678165 -0.36339998 -2.82600681 -3.94488929

```

```

## 1989 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 1990 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 1991 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 1992 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 1993 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 1994 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 1995 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 1996 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 1997 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 1998 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 1999 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2000 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2001 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2002 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2003 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2004 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2005 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2006 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2007 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2008 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2009 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2010 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2011 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2012 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2013 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2014 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2015 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2016 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2017 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2018 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2019 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2020 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2021 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2022 3.21594084 2.64254447 1.13678165 -0.36339998 -2.82600681 -3.94488929
## 2023 3.21594084 2.64254447 1.13678165

```

```
##
```

```
## $trend
```

```

##          Jan          Feb          Mar          Apr          May
## 1984          NA          NA          NA          NA          NA
## 1985 0.002291667 0.002583333 0.002833333 0.002958333 0.003000000
## 1986 0.003750000 0.003791667 0.003750000 0.003791667 0.003916667
## 1987 0.003666667 0.003583333 0.003416667 0.003250000 0.003083333
## 1988 0.003000000 0.002916667 0.002916667 0.002833333 0.002708333
## 1989 2.314416667 2.766791667 3.196916667 3.590208333 3.931791667
## 1990 4.533500000 4.564666667 4.592958333 4.616625000 4.636125000
## 1991 4.748375000 4.771500000 4.792875000 4.810041667 4.822666667
## 1992 4.915291667 4.928250000 4.941208333 4.954583333 4.968458333
## 1993 5.070541667 5.085833333 5.101916667 5.115166667 5.125083333
## 1994 5.225791667 5.234500000 5.242875000 5.257291667 5.271166667
## 1995 5.323666667 5.339041667 5.351583333 5.356208333 5.359166667
## 1996 5.422750000 5.422000000 5.415208333 5.412375000 5.414250000
## 1997 5.367041667 5.364000000 5.364500000 5.362208333 5.356875000
## 1998 5.302000000 5.299083333 5.297833333 5.296000000 5.292791667
## 1999 5.228375000 5.211541667 5.195375000 5.184333333 5.177416667

```

##	2000	5.041958333	5.023041667	5.006375000	4.988791667	4.969666667
##	2001	4.849666667	4.831791667	4.814083333	4.792791667	4.772750000
##	2002	4.668125000	4.649583333	4.631000000	4.616666667	4.607916667
##	2003	4.531208333	4.513291667	4.499750000	4.489750000	4.478750000
##	2004	4.437416667	4.435833333	4.432416667	4.427708333	4.423166667
##	2005	4.368708333	4.359166667	4.351500000	4.344708333	4.338500000
##	2006	4.420083333	4.439375000	4.458208333	4.473583333	4.488125000
##	2007	4.634541667	4.658291667	4.679583333	4.702625000	4.722416667
##	2008	4.935375000	4.971500000	5.006125000	5.035666667	5.059125000
##	2009	5.146583333	5.161125000	5.174000000	5.185458333	5.196875000
##	2010	5.440416667	5.488208333	5.534541667	5.573166667	5.606791667
##	2011	6.008125000	6.076083333	6.144041667	6.209000000	6.265333333
##	2012	7.060458333	7.217041667	7.363250000	7.508625000	7.640416667
##	2013	8.829250000	9.027375000	9.245791667	9.467583333	9.677791667
##	2014	11.881458333	12.241666667	12.590333333	12.908708333	13.172791667
##	2015	15.178125000	15.526500000	15.796958333	15.956375000	16.091958333
##	2016	18.498750000	19.016125000	19.524208333	20.010583333	20.421291667
##	2017	24.926625000	25.523708333	26.083416667	26.637000000	27.017000000
##	2018	30.275250000	30.790791667	31.239416667	31.523500000	31.744791667
##	2019	33.740375000	34.281458333	34.736916667	35.159416667	35.513750000
##	2020	39.655041667	40.379416667	40.861666667	41.266916667	41.741250000
##	2021	47.830333333	48.648125000	49.715208333	50.681750000	51.436666667
##	2022	59.136541667	60.355500000	61.356291667	62.376750000	63.128333333
##	2023	67.544625000	68.824291667	69.965500000	NA	NA
##		Jun	Jul	Aug	Sep	Oct
##	1984	NA	0.001500000	0.001583333	0.001541667	0.001625000
##	1985	0.003041667	0.003083333	0.003166667	0.003375000	0.003541667
##	1986	0.004000000	0.003916667	0.003833333	0.003833333	0.003791667
##	1987	0.003000000	0.003083333	0.003208333	0.003291667	0.003250000
##	1988	0.002625000	0.120041667	0.364083333	0.666333333	1.036083333
##	1989	4.230208333	4.382666667	4.399708333	4.419875000	4.443000000
##	1990	4.653708333	4.668083333	4.677375000	4.685083333	4.691416667
##	1991	4.832750000	4.839541667	4.845916667	4.857291667	4.876000000
##	1992	4.979916667	4.988833333	4.997625000	5.011000000	5.029041667
##	1993	5.134875000	5.146083333	5.156666667	5.168875000	5.181708333
##	1994	5.280125000	5.283458333	5.283250000	5.282416667	5.285041667
##	1995	5.363916667	5.368625000	5.373833333	5.384500000	5.397750000
##	1996	5.418083333	5.418916667	5.416916667	5.412125000	5.402375000
##	1997	5.350791667	5.345708333	5.338208333	5.331083333	5.326000000
##	1998	5.289666667	5.284625000	5.278791667	5.269916667	5.258166667
##	1999	5.170708333	5.161625000	5.150416667	5.134541667	5.111708333
##	2000	4.953416667	4.939541667	4.925958333	4.910541667	4.893458333
##	2001	4.755875000	4.743708333	4.737166667	4.729708333	4.719291667
##	2002	4.599250000	4.590958333	4.583416667	4.574791667	4.567166667
##	2003	4.468291667	4.462541667	4.458250000	4.453708333	4.448791667
##	2004	4.419916667	4.415708333	4.410791667	4.403208333	4.394250000
##	2005	4.331791667	4.334125000	4.346416667	4.359958333	4.374625000
##	2006	4.501583333	4.513541667	4.525208333	4.541750000	4.561833333
##	2007	4.738208333	4.753416667	4.771708333	4.796833333	4.829375000
##	2008	5.080333333	5.093041667	5.097250000	5.104750000	5.115416667
##	2009	5.206458333	5.220416667	5.240958333	5.265458333	5.296791667
##	2010	5.639375000	5.672708333	5.713541667	5.763458333	5.821125000
##	2011	6.315750000	6.372208333	6.429958333	6.505291667	6.607000000
##	2012	7.758708333	7.868916667	7.987583333	8.141666667	8.315291667

```

## 2013 9.872541667 10.066250000 10.250583333 10.479750000 10.777541667
## 2014 13.354625000 13.489833333 13.670625000 13.922541667 14.234583333
## 2015 16.257916667 16.427625000 16.671833333 16.974166667 17.251291667
## 2016 20.755125000 21.066875000 21.327541667 21.769375000 22.475583333
## 2017 27.253500000 27.587750000 28.013500000 28.380250000 28.768583333
## 2018 31.919000000 32.045916667 32.152625000 32.373500000 32.734708333
## 2019 35.728875000 35.999250000 36.524208333 36.997041667 37.377875000
## 2020 42.309541667 42.830291667 43.197333333 43.874916667 44.948625000
## 2021 52.025458333 52.645916667 53.548208333 54.515291667 55.487500000
## 2022 63.518083333 63.824416667 64.061958333 64.377916667 64.911833333
## 2023      NA      NA      NA      NA
##      Nov      Dec
## 1984 0.001875000 0.002083333
## 1985 0.003625000 0.003666667
## 1986 0.003666667 0.003625000
## 1987 0.003166667 0.003125000
## 1988 1.442875000 1.871000000
## 1989 4.471833333 4.503708333
## 1990 4.704083333 4.724625000
## 1991 4.892750000 4.904083333
## 1992 5.045750000 5.058125000
## 1993 5.193708333 5.210250000
## 1994 5.295416667 5.309291667
## 1995 5.408041667 5.416708333
## 1996 5.389375000 5.375875000
## 1997 5.317541667 5.308166667
## 1998 5.250666667 5.242541667
## 1999 5.087291667 5.063208333
## 2000 4.877666667 4.865166667
## 2001 4.702333333 4.684250000
## 2002 4.559166667 4.547750000
## 2003 4.444916667 4.440833333
## 2004 4.386583333 4.378416667
## 2005 4.390166667 4.404875000
## 2006 4.583500000 4.607958333
## 2007 4.862625000 4.897833333
## 2008 5.128208333 5.137333333
## 2009 5.337583333 5.388666667
## 2010 5.882083333 5.944916667
## 2011 6.738500000 6.895083333
## 2012 8.482083333 8.652416667
## 2013 11.122416667 11.504791667
## 2014 14.562750000 14.859750000
## 2015 17.596791667 18.017625000
## 2016 23.262333333 24.147166667
## 2017 29.251250000 29.755541667
## 2018 33.024958333 33.299583333
## 2019 38.084583333 38.879708333
## 2020 46.005875000 47.006458333
## 2021 56.525166667 57.759708333
## 2022 65.700500000 66.535208333
## 2023
##
## $random

```

##	Jan	Feb	Mar	Apr	May
## 1984	NA	NA	NA	NA	NA
## 1985	4.003914933	3.154645702	-0.035158785	-1.401495998	-2.801091393
## 1986	4.002456600	3.155437369	-0.033075452	-1.401329332	-2.801008060
## 1987	4.000539933	3.155645702	-0.032742118	-1.401787665	-2.802174726
## 1988	4.003206600	3.157312369	-0.031242118	-1.403370998	-2.801799726
## 1989	4.510789933	3.431437369	0.990757882	-0.332745998	-1.625883060
## 1990	2.485706600	1.848562369	-0.136283785	-1.073162665	-1.924216393
## 1991	2.380831600	1.754729035	-0.264200452	-1.186579332	-1.886758060
## 1992	2.284914933	1.679979035	-0.221533785	-1.073120998	-1.888549726
## 1993	2.221664933	1.641395702	-0.180242118	-1.002704332	-1.875174726
## 1994	2.210414933	1.602729035	-0.138200452	-1.019829332	-1.858258060
## 1995	2.095539933	1.510187369	-0.278908785	-1.023745998	-1.792258060
## 1996	2.053456600	1.495229035	-0.154533785	-0.949912665	-1.730341393
## 1997	2.059164933	1.555229035	-0.220825452	-1.016745998	-1.867966393
## 1998	2.077206600	1.487145702	-0.192158785	-1.034537665	-1.922883060
## 1999	2.069831600	1.515687369	-0.243700452	-1.050870998	-1.859508060
## 2000	2.117248266	1.574187369	-0.305700452	-1.152329332	-1.940758060
## 2001	2.162539933	1.586437369	-0.304408785	-1.175329332	-1.903841393
## 2002	2.259081600	1.696645702	-0.228325452	-1.142204332	-2.003008060
## 2003	2.320998266	1.726937369	-0.198075452	-1.097287665	-1.983841393
## 2004	2.375789933	1.740395702	-0.175742118	-1.108245998	-1.948258060
## 2005	2.377498266	1.766062369	-0.225825452	-1.109245998	-1.963591393
## 2006	2.465123266	1.841854035	-0.163533785	-1.055120998	-1.923216393
## 2007	2.385664933	1.767937369	-0.132908785	-1.054162665	-1.867508060
## 2008	2.274831600	1.703729035	-0.105450452	-0.960204332	-1.833216393
## 2009	2.104623266	1.574104035	-0.153325452	-0.973995998	-1.799966393
## 2010	2.048789933	1.502020702	-0.180867118	-0.942704332	-1.649883060
## 2011	1.920081600	1.455145702	-0.133367118	-0.851537665	-1.572424726
## 2012	1.550748266	1.017187369	-0.247575452	-0.815162665	-1.127508060
## 2013	1.043956600	0.792854035	-0.018117118	-0.719120998	-1.216883060
## 2014	0.279748266	-0.285437631	0.001341215	-0.377245998	-0.217883060
## 2015	-1.358918400	-0.889270965	0.159716215	0.699087335	0.614950274
## 2016	-2.766543400	-0.430895965	-0.259533785	-0.012120998	1.234616940
## 2017	-5.367418400	-4.509479298	1.356257882	2.135462335	4.746908607
## 2018	-5.854043400	-4.420562631	-0.353742118	3.122962335	5.729116940
## 2019	-7.487168400	-7.182229298	0.720757882	3.584045668	4.829158607
## 2020	-8.909835067	-5.173187631	-2.162992118	3.375545668	9.663658607
## 2021	-11.792126734	-9.925895965	1.729466215	6.983712335	12.319241940
## 2022	-13.324335067	-9.752270965	1.417382882	7.292712335	13.527575274
## 2023	-19.108418400	-15.144062631	-2.685825452	NA	NA
##	Jun	Jul	Aug	Sep	Oct
## 1984	NA	-3.216440836	-2.641127802	-1.135323315	0.363774976
## 1985	-3.060551919	-3.215024170	-2.638711135	-1.135156648	0.362858309
## 1986	-3.061510253	-3.213857503	-2.640377802	-1.135614982	0.363608309
## 1987	-3.059510253	-3.213024170	-2.641752802	-1.137073315	0.362149976
## 1988	-3.060135253	-3.331982503	-3.002627802	-1.800114982	-0.672683358
## 1989	-2.117718586	-2.122607503	-1.653252802	-0.615656648	0.421399976
## 1990	-2.182218586	-2.052024170	-1.538919469	-0.593864982	0.453983309
## 1991	-2.092260253	-1.922482503	-1.453461135	-0.507073315	0.422399976
## 1992	-2.111426919	-1.930774170	-1.435169469	-0.519781648	0.449358309
## 1993	-2.139385253	-1.917024170	-1.398211135	-0.487656648	0.424691642
## 1994	-2.050635253	-1.915399170	-1.454794469	-0.470198315	0.536358309
## 1995	-1.955426919	-1.834565836	-1.342377802	-0.474281648	0.436649976

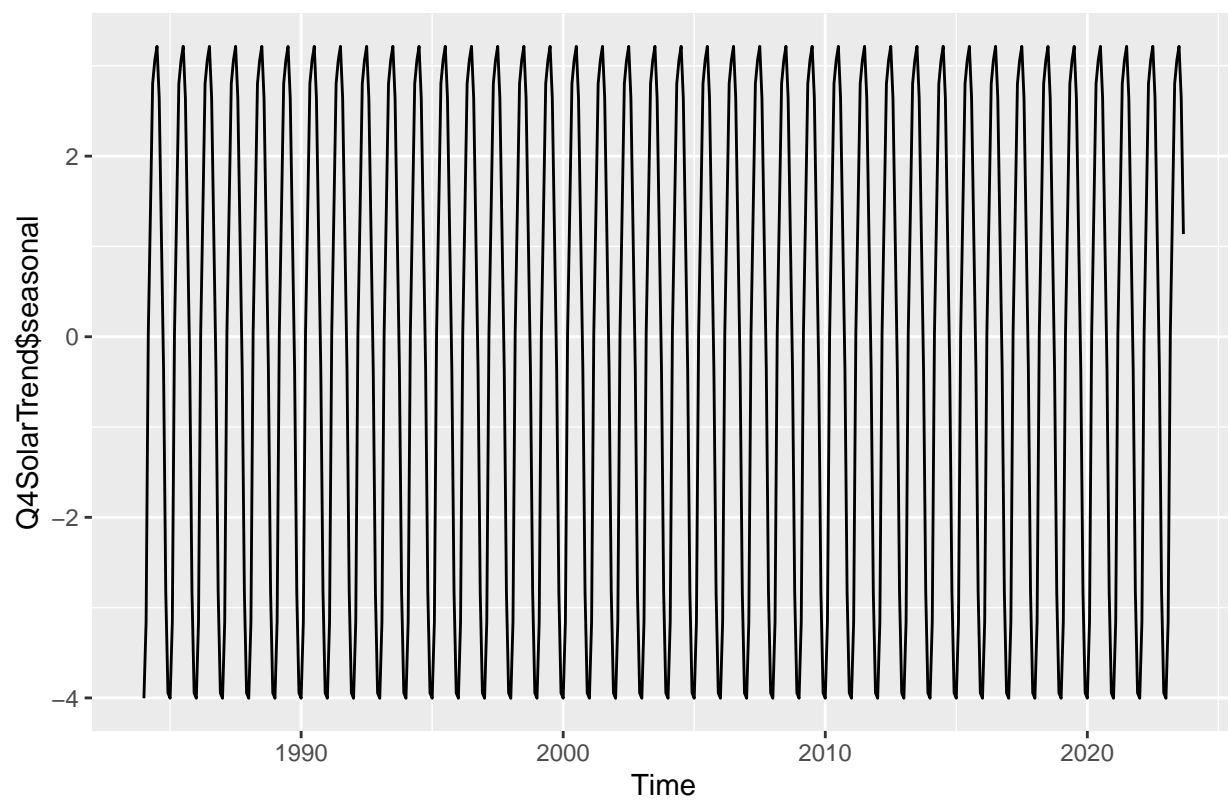


## 1996	-1.918593586	-1.830857503	-1.457461135	-0.592906648	0.455024976
## 1997	-1.980301919	-1.840649170	-1.368752802	-0.509864982	0.474399976
## 1998	-2.025176919	-1.821565836	-1.337336135	-0.450698315	0.500233309
## 1999	-2.049218586	-1.895565836	-1.415961135	-0.496323315	0.562691642
## 2000	-2.120926919	-1.894482503	-1.424502802	-0.439323315	0.525941642
## 2001	-2.063385253	-1.930649170	-1.432711135	-0.486489982	0.417108309
## 2002	-2.076760253	-1.994899170	-1.506961135	-0.549573315	0.443233309
## 2003	-2.109801919	-2.099482503	-1.578794469	-0.556489982	0.449608309
## 2004	-2.139426919	-2.056649170	-1.565336135	-0.553989982	0.439149976
## 2005	-2.147301919	-2.112065836	-1.592961135	-0.602739982	0.387774976
## 2006	-2.154093586	-2.089482503	-1.510752802	-0.593531648	0.378566642
## 2007	-2.093718586	-1.988357503	-1.528252802	-0.566614982	0.382024976
## 2008	-1.961843586	-1.900982503	-1.413794469	-0.483531648	0.413983309
## 2009	-2.039968586	-1.854357503	-1.382502802	-0.510239982	0.373608309
## 2010	-1.806885253	-1.730649170	-1.284086135	-0.467239982	0.235274976
## 2011	-1.711260253	-1.685149170	-1.114502802	-0.464073315	0.263399976
## 2012	-1.216218586	-1.150857503	-0.945127802	-0.318448315	0.262108309
## 2013	-1.190051919	-1.244190836	-0.557127802	-0.065531648	0.531858309
## 2014	0.010864747	-0.310774170	0.310830531	0.571676685	0.635816642
## 2015	0.483573081	1.016434164	1.405622198	-0.084948315	-0.949891691
## 2016	1.137364747	2.773184164	2.770913865	1.292843352	-0.674183358
## 2017	5.766989747	3.831309164	2.835955531	1.363968352	-0.363183358
## 2018	7.494489747	5.453142497	4.989830531	1.844718352	-1.985308358
## 2019	7.406614747	8.356809164	6.747247198	2.023176685	-1.290475024
## 2020	8.846948081	12.112767497	6.872122198	-0.078698315	-3.911225024
## 2021	10.794031414	10.407142497	8.038247198	3.373926685	-5.346100024
## 2022	16.030406414	15.543642497	10.464497198	4.590301685	-1.358433358
## 2023	NA	NA	NA	NA	
##	Nov	Dec			
## 1984	2.825131813	3.942805959			
## 1985	2.823381813	3.942222625			
## 1986	2.825340147	3.942264292			
## 1987	2.823840147	3.942764292			
## 1988	1.383131813	2.073889292			
## 1989	2.051173480	2.906180959			
## 1990	2.005923480	2.920264292			
## 1991	1.967256813	2.832805959			
## 1992	1.967256813	2.800764292			
## 1993	1.929298480	2.773639292			
## 1994	1.945590147	2.771597625			
## 1995	1.890965147	2.720180959			
## 1996	1.931631813	2.831014292			
## 1997	1.932465147	2.823722625			
## 1998	1.964340147	2.849347625			
## 1999	2.045715147	2.949680959			
## 2000	2.051340147	2.961722625			
## 2001	2.028673480	2.935639292			
## 2002	2.087840147	2.948139292			
## 2003	2.050090147	2.956055959			
## 2004	2.064423480	2.984472625			
## 2005	1.982840147	2.875014292			
## 2006	1.960506813	2.823930959			
## 2007	1.885381813	2.709055959			
## 2008	1.864798480	2.733555959			

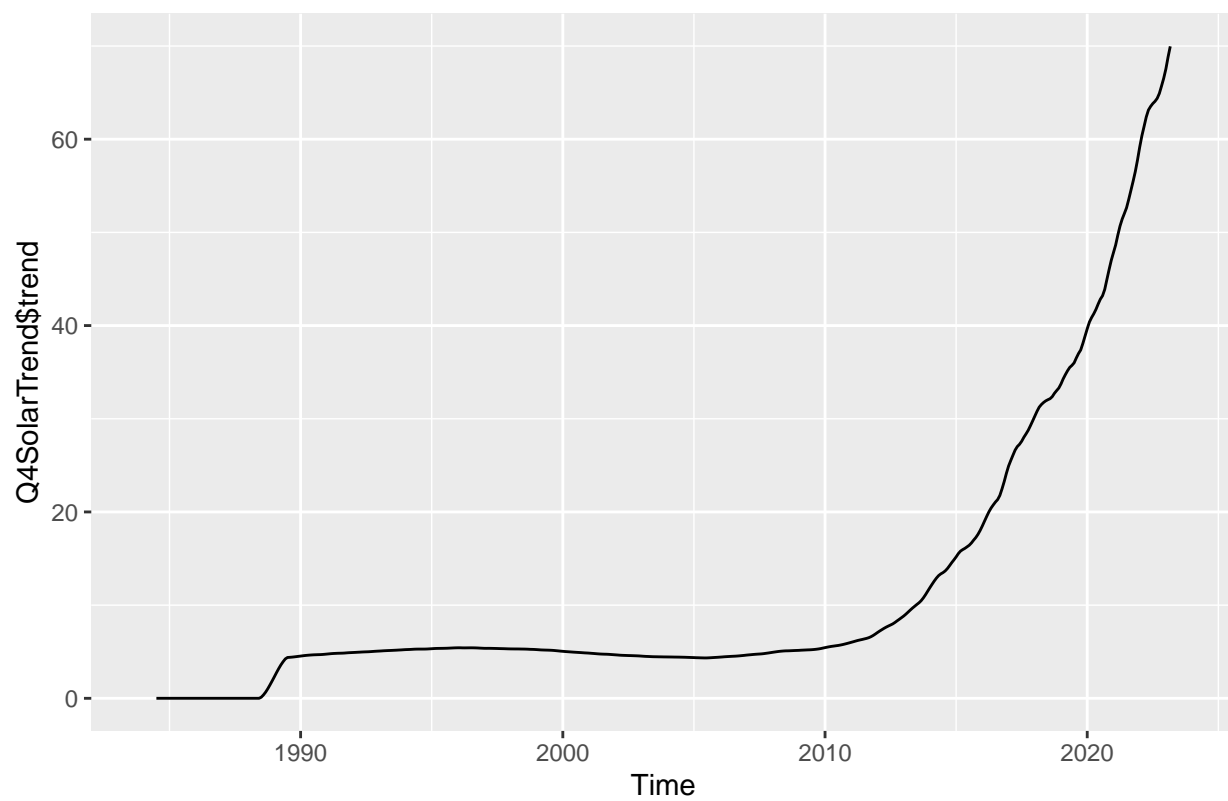
```

## 2009    1.788423480    2.579222625
## 2010    1.664923480    2.383972625
## 2011    1.346506813    2.105805959
## 2012    1.058923480    1.731472625
## 2013    0.731590147    1.240097625
## 2014    0.068256813   -0.527860708
## 2015   -1.142784853   -1.525735708
## 2016   -2.451326520   -4.000277375
## 2017   -5.924243187   -6.448652375
## 2018   -6.730951520   -8.778694041
## 2019   -8.624576520  -12.361819041
## 2020  -10.111868187  -13.283569041
## 2021  -11.617159853  -18.919819041
## 2022  -16.166493187  -22.967319041
## 2023
##
## $figure
## [1] -4.00420660 -3.15722904  0.03232545  1.40253766  2.80309139  3.06251025
## [7]  3.21594084  2.64254447  1.13678165 -0.36339998 -2.82600681 -3.94488929
##
## $type
## [1] "additive"
##
## attr(,"class")
## [1] "decomposed.ts"
autoplot(Q4SolarTrend$seasonal)

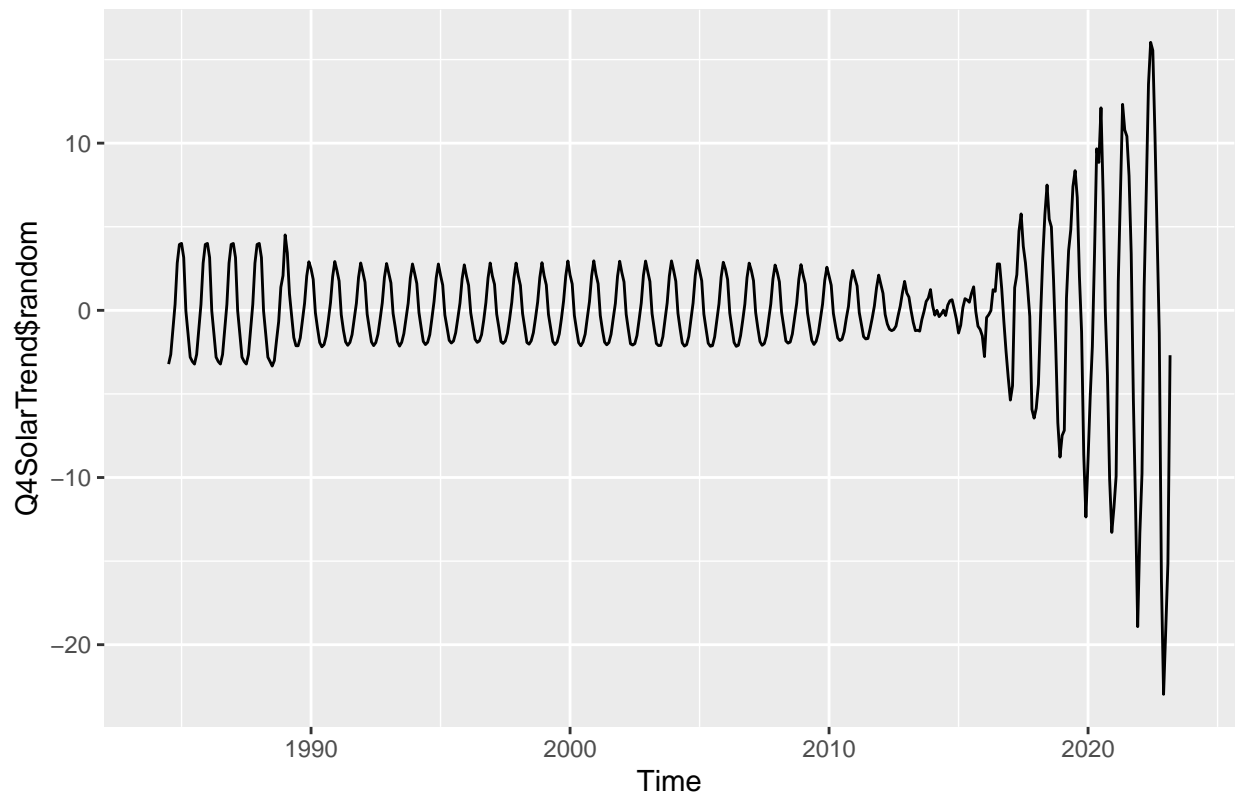
```



```
autoplot(Q4SolarTrend$trend)
```



```
autoplot(Q4SolarTrend$random)
```



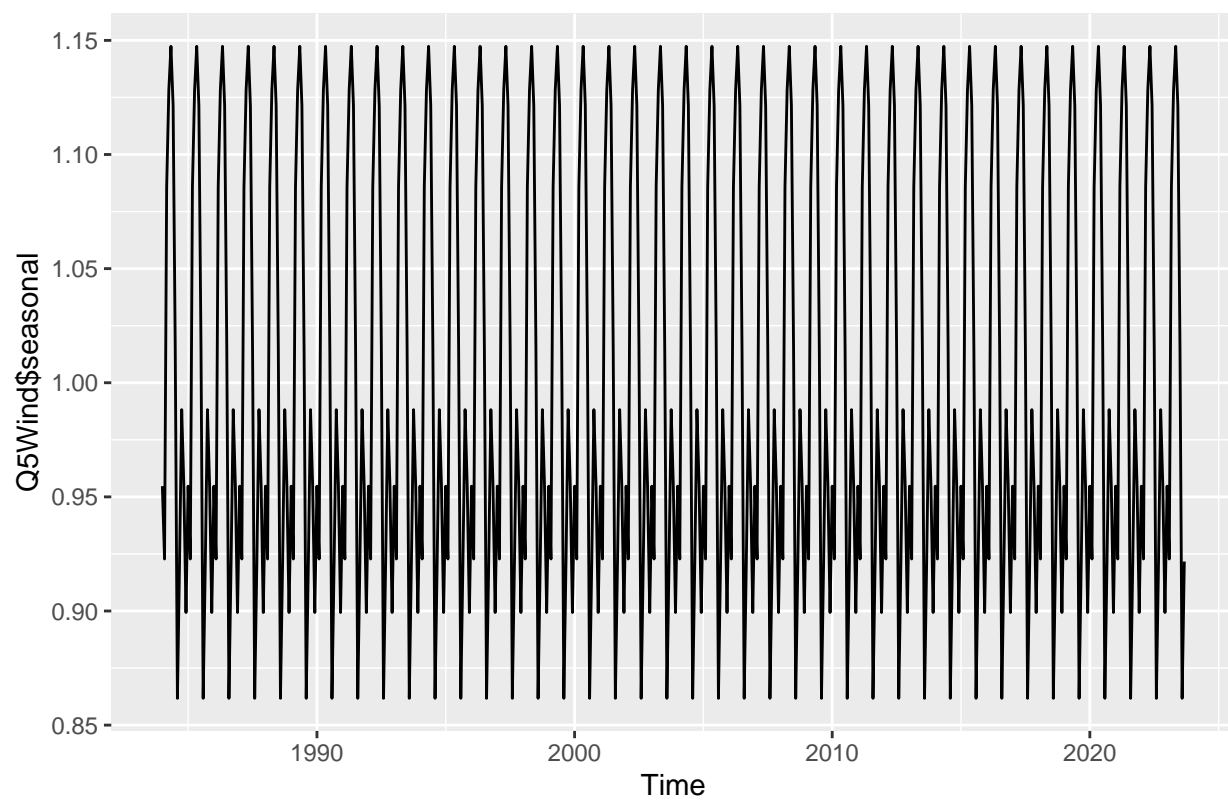
### Q5

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?

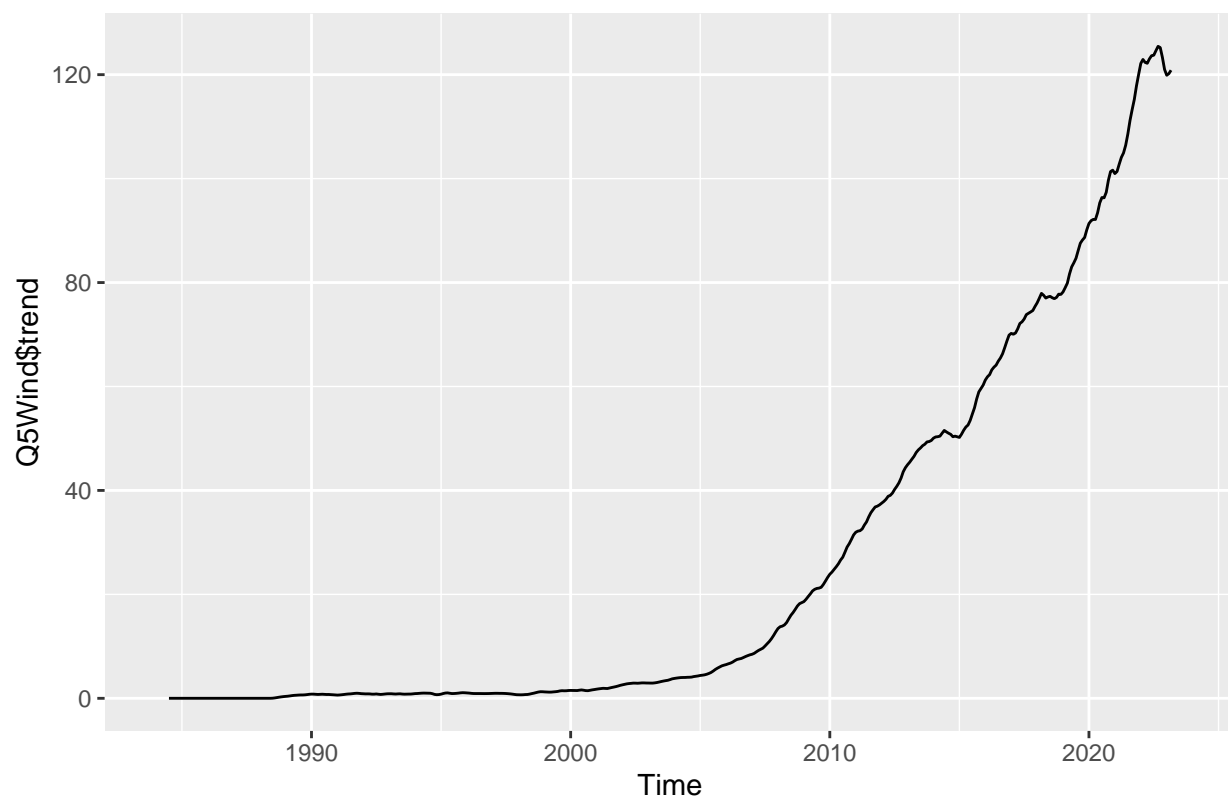
Answer: The random component looks better this time (expecially for wind) as there is more randomness to the plot and hence less evidence of seasonaility. There is still some indication that the timeframe should be fractionated for models.

```
Q5Wind <- decompose(WindTS, type = "multiplicative")
```

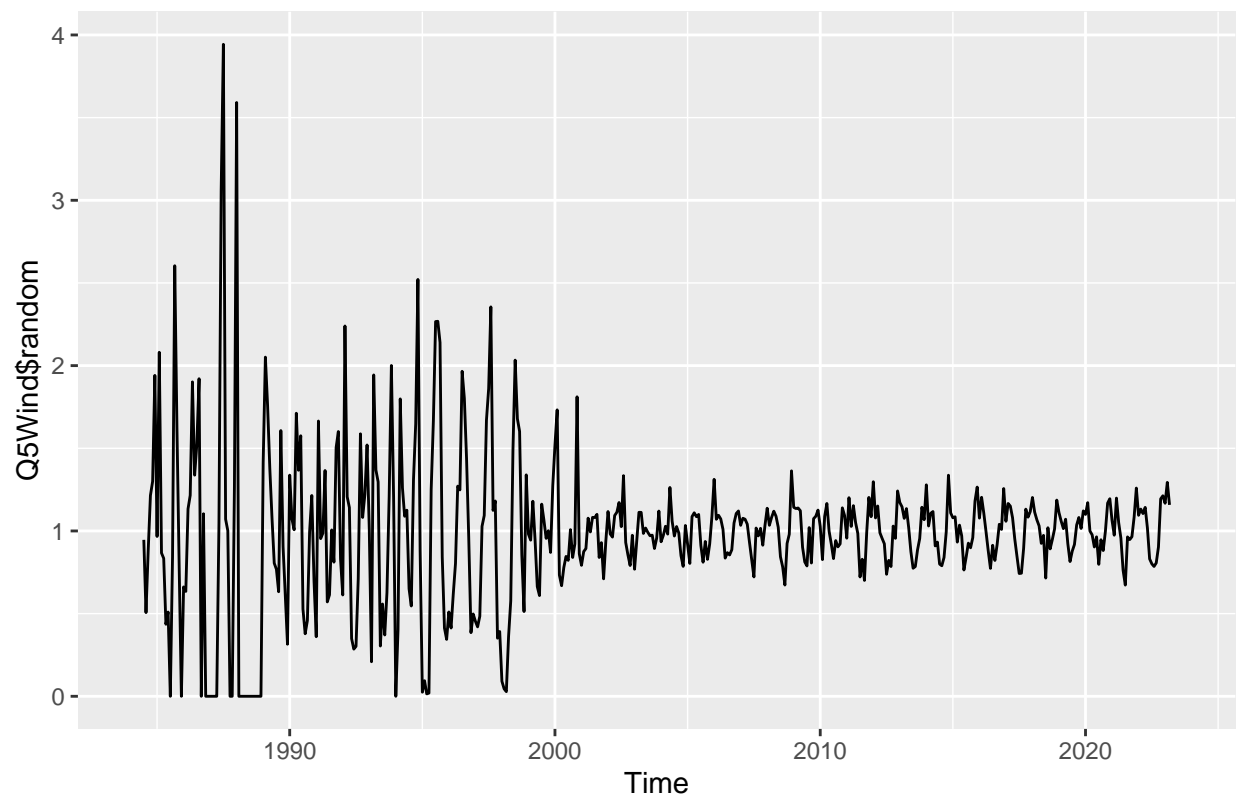
```
autoplot(Q5Wind$seasonal)
```



```
autoplot(Q5Wind$trend)
```



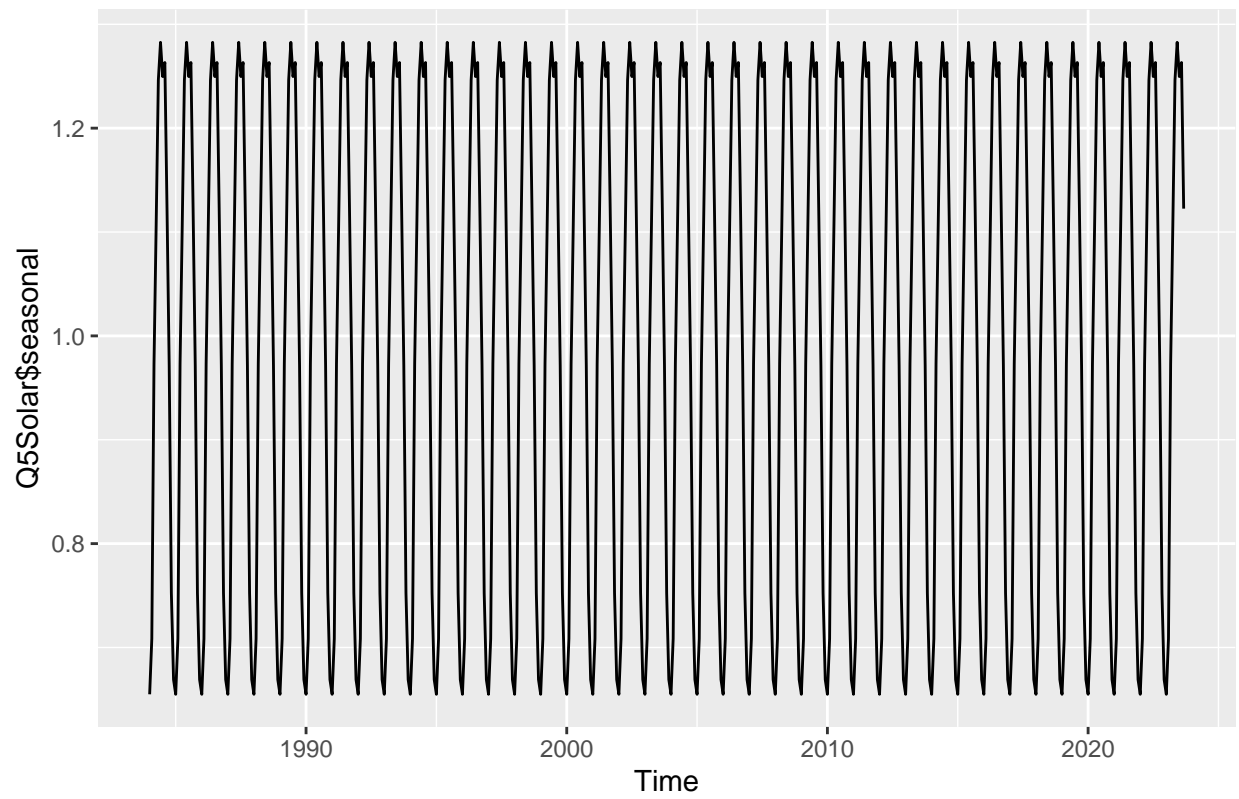
```
autoplot(Q5Wind$random)
```



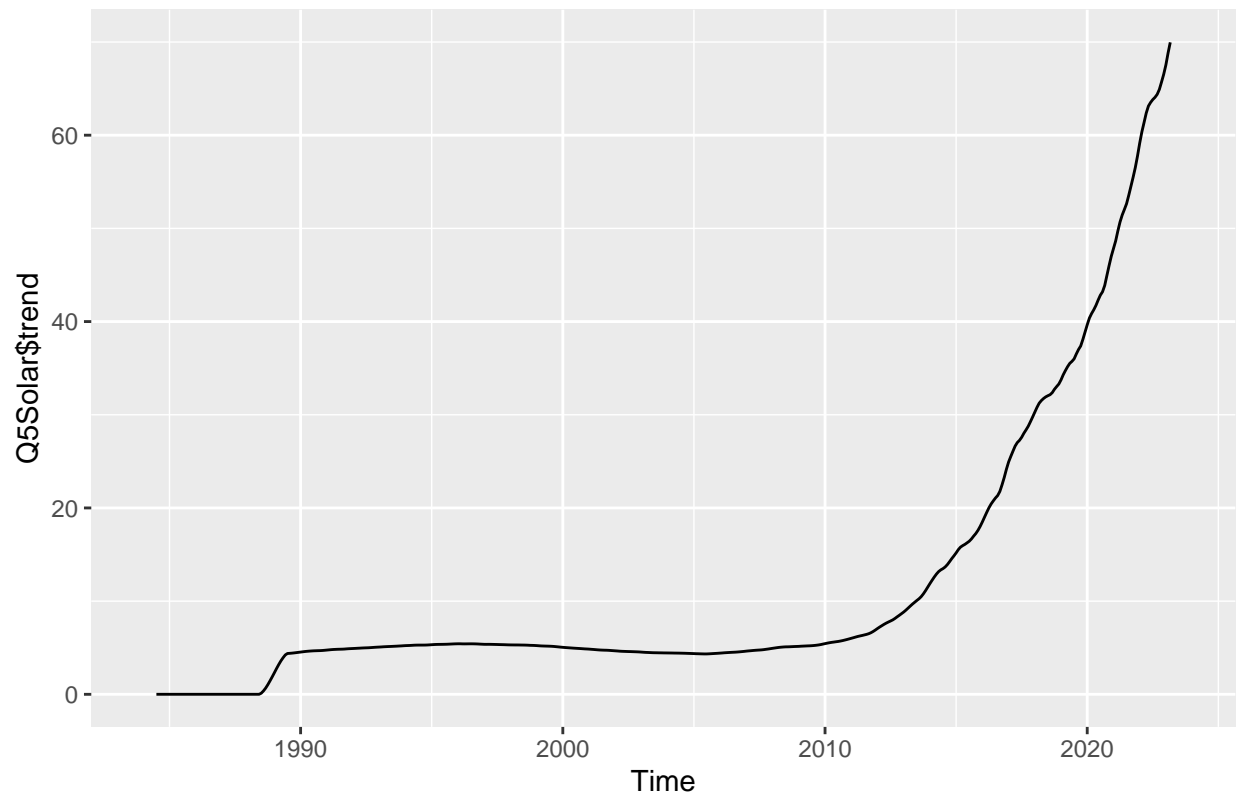
```
Q5Solar <- decompose(SolarTS, type = "multiplicative")
```

```
autoplot(Q5Solar$seasonal)
```

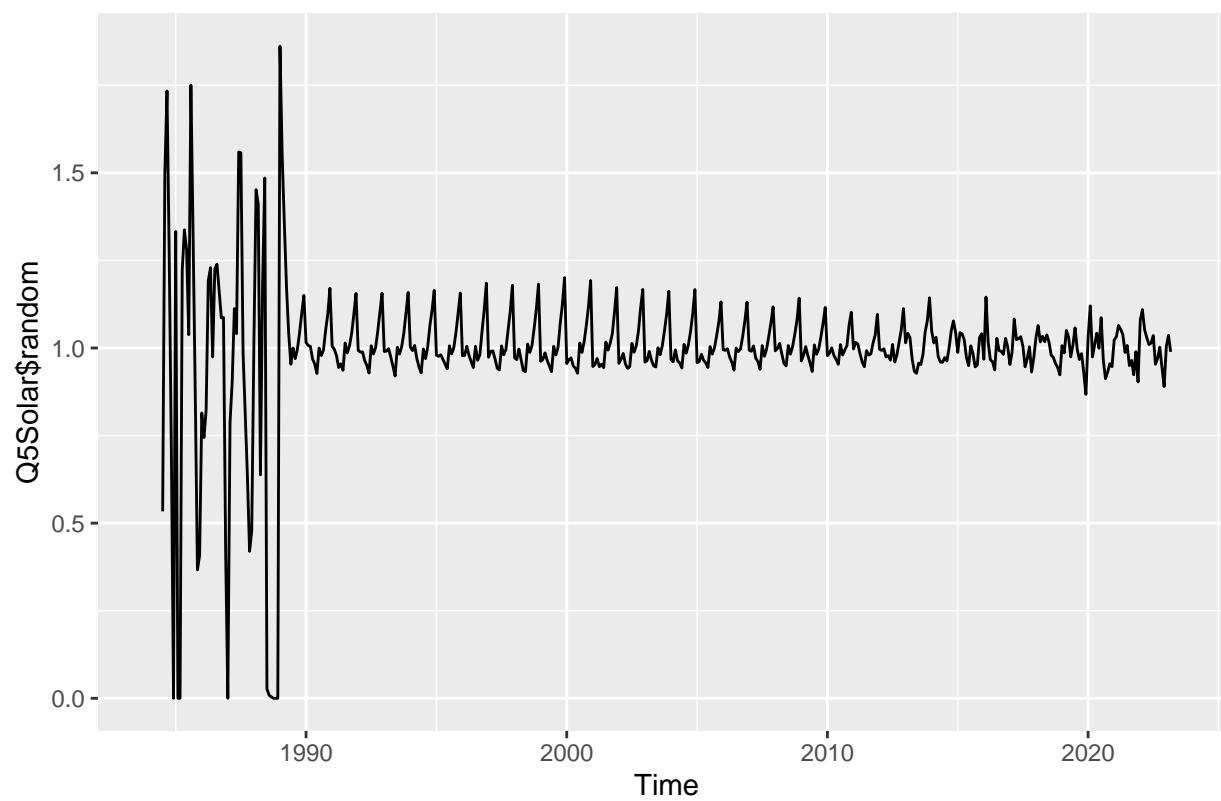




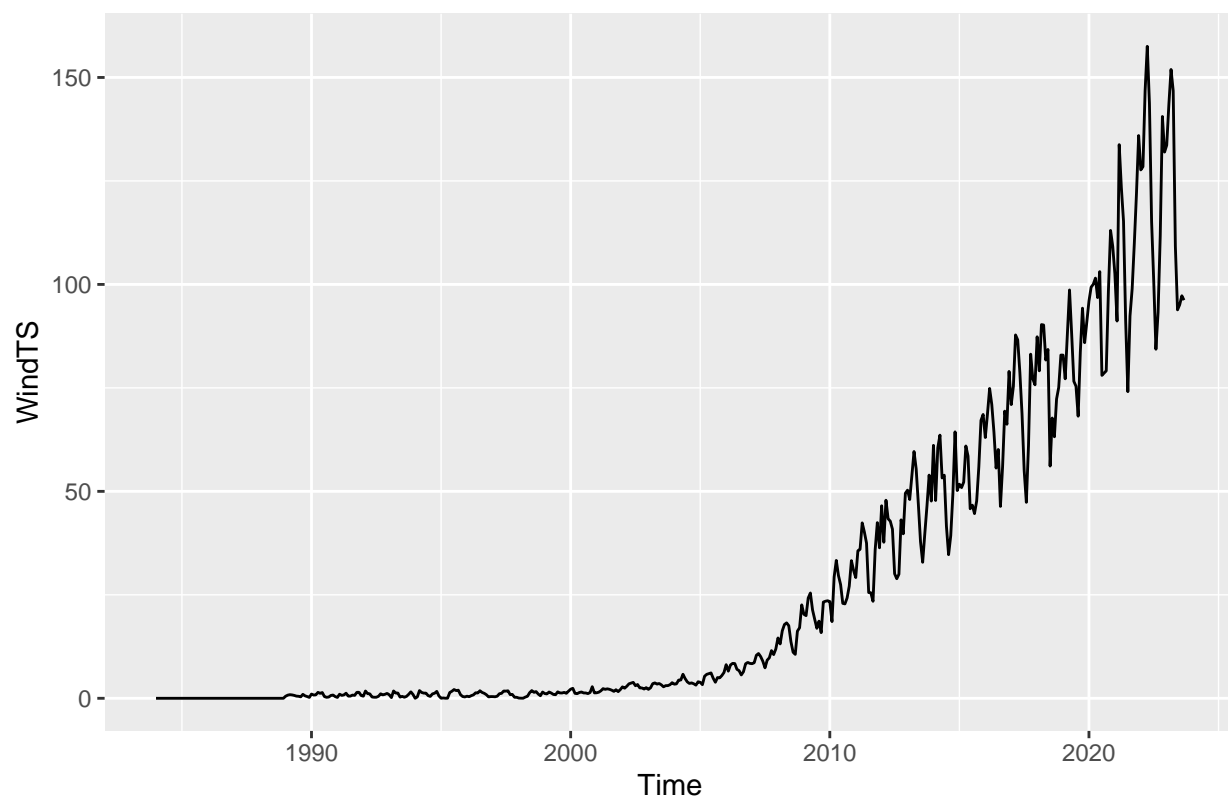
```
autoplot(Q5Solar$trend)
```



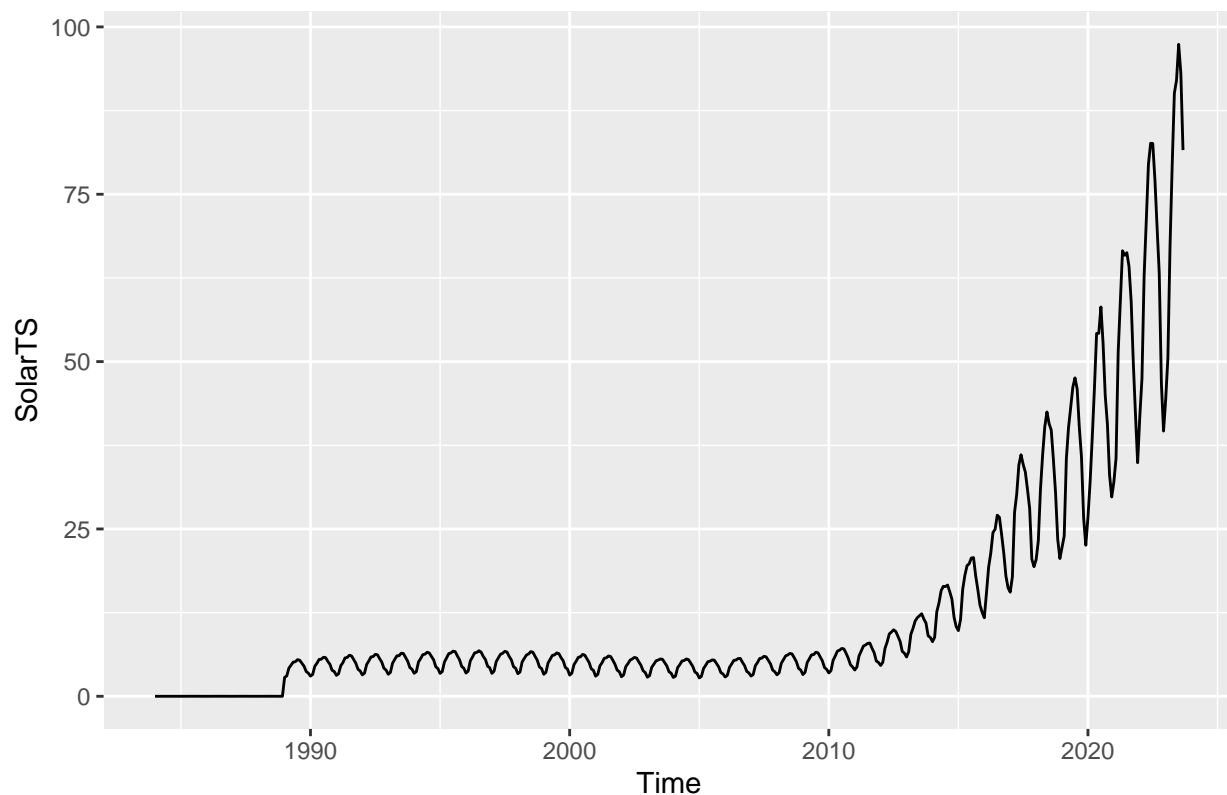
```
autoplot(Q5Solar$random)
```



```
autoplot(WindTS)
```



```
autoplot(SolarTS)
```



### 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: No, I do not think all of the historic data is necessary. There have been many factors over time like increased environmental awareness, technology improvements as well as cost reductions in solar and wind technology all of which have fundamentally changed the way we consume solar and wind energy. Hence, it is probably better to look at the last year or two to predict the next six months as the technology from 1 or 2 years ago is far similar to the systems we are using today than information/technology from 30 years ago.

### 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.

```
FilterData <- energy_data |>
  filter(year(as.Date(Month, "%Y")) >= 2012)
```

```
FilterDataTS <- ts(FilterData)
```

```
FilterWind <- FilterData |>
  select(`Wind Energy Consumption`)
```

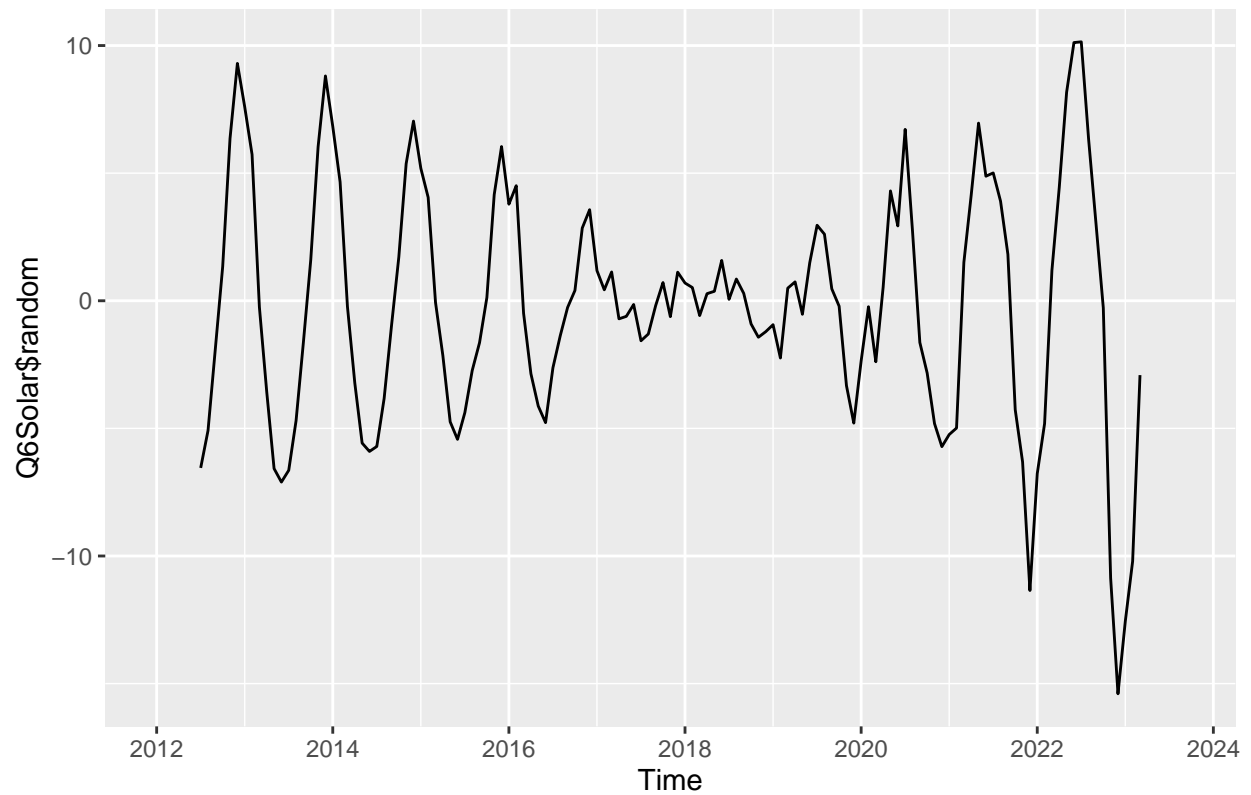
```
FilterSolar <- FilterData |>  
  select(`Solar Energy Consumption`)
```

```
TSFilterWind <- ts(FilterWind, start = c(2012, 1), frequency = 12)  
TSFilterSolar <- ts(FilterSolar, start = c(2012, 1), frequency = 12)
```

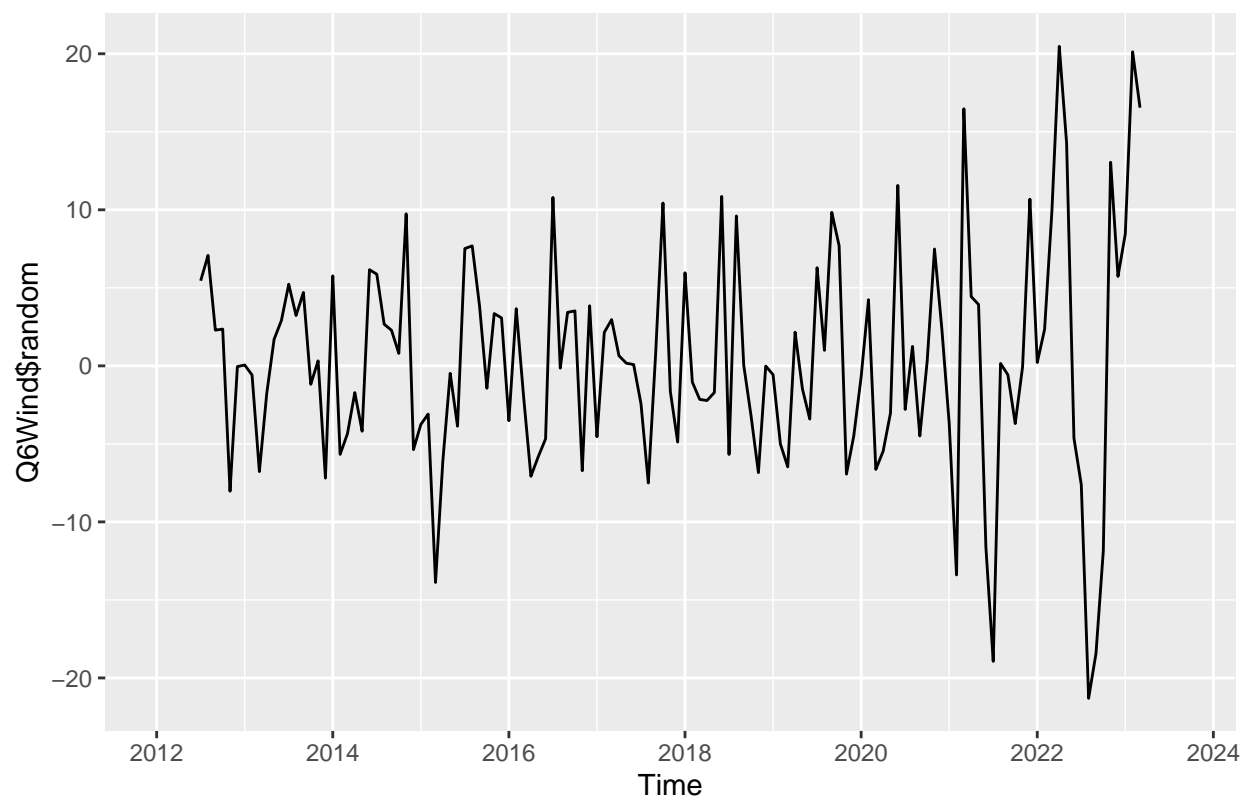
```
Q6Wind <- decompose(TSFilterWind, type = "additive")
```

```
Q6Solar <- decompose(TSFilterSolar, type = "additive")
```

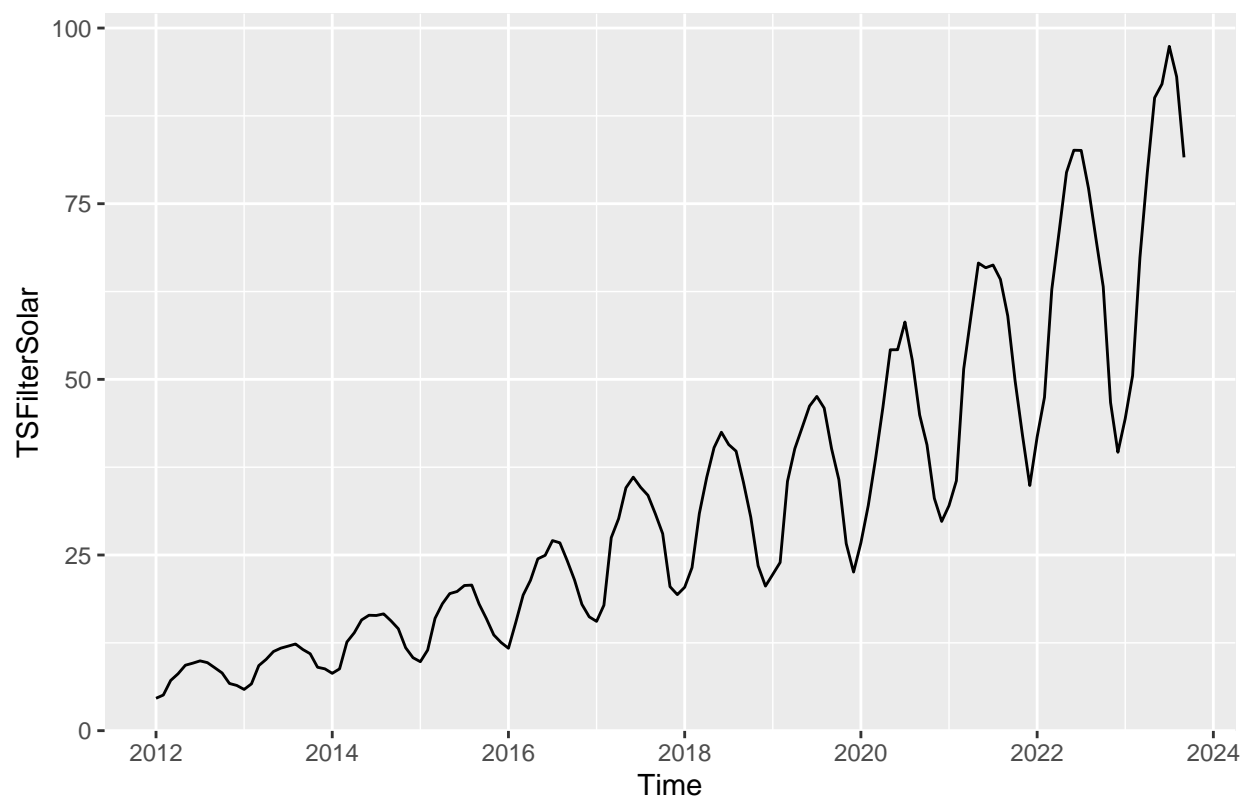
```
autoplot(Q6Solar$random)
```



```
autoplot(Q6Wind$random)
```

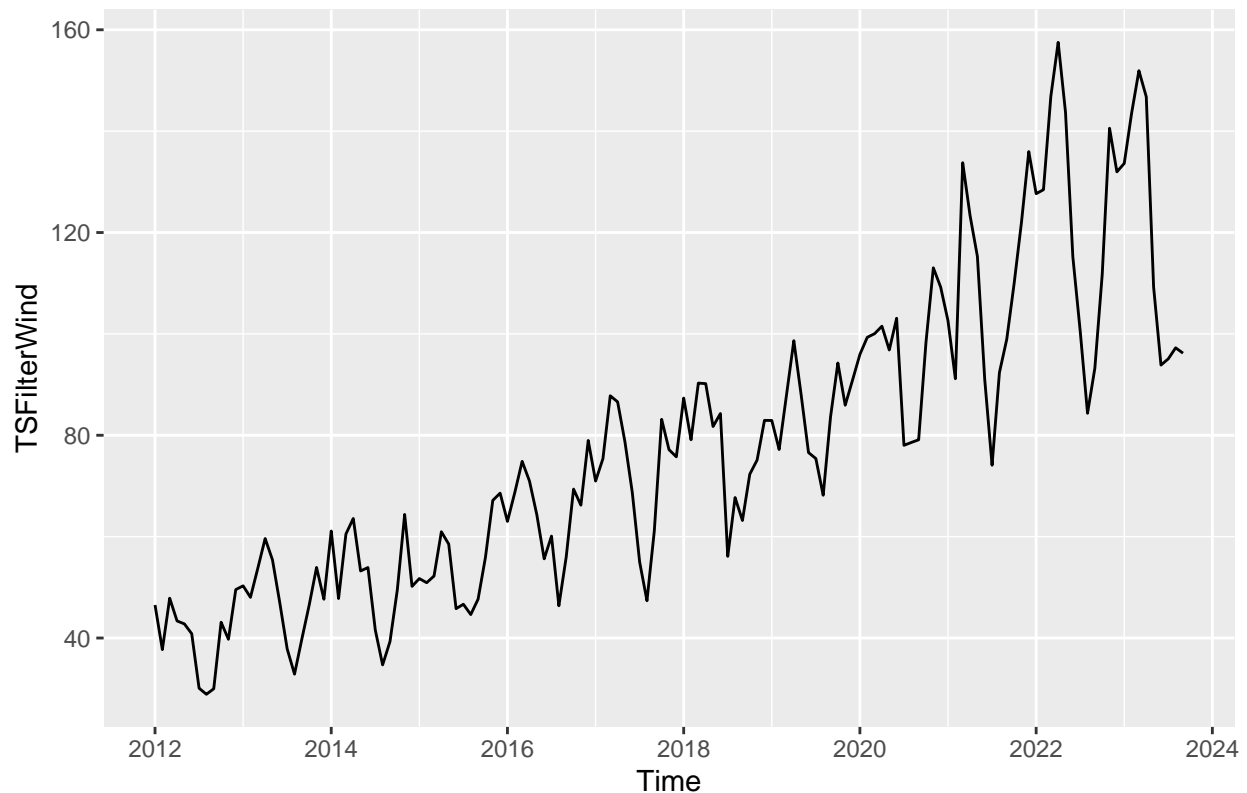


```
autoplot(TSFilterSolar)
```



```
autoplot(TSFilterWind)
```





Answer: The random components do look much better here and are far more random; there are no distinguishable patterns that emerge year to year. Of course, this will never be completely random as we expect some noise in every data set but this version of the data is the best explained by the model so far.

## Identify and Remove outliers

### Q8

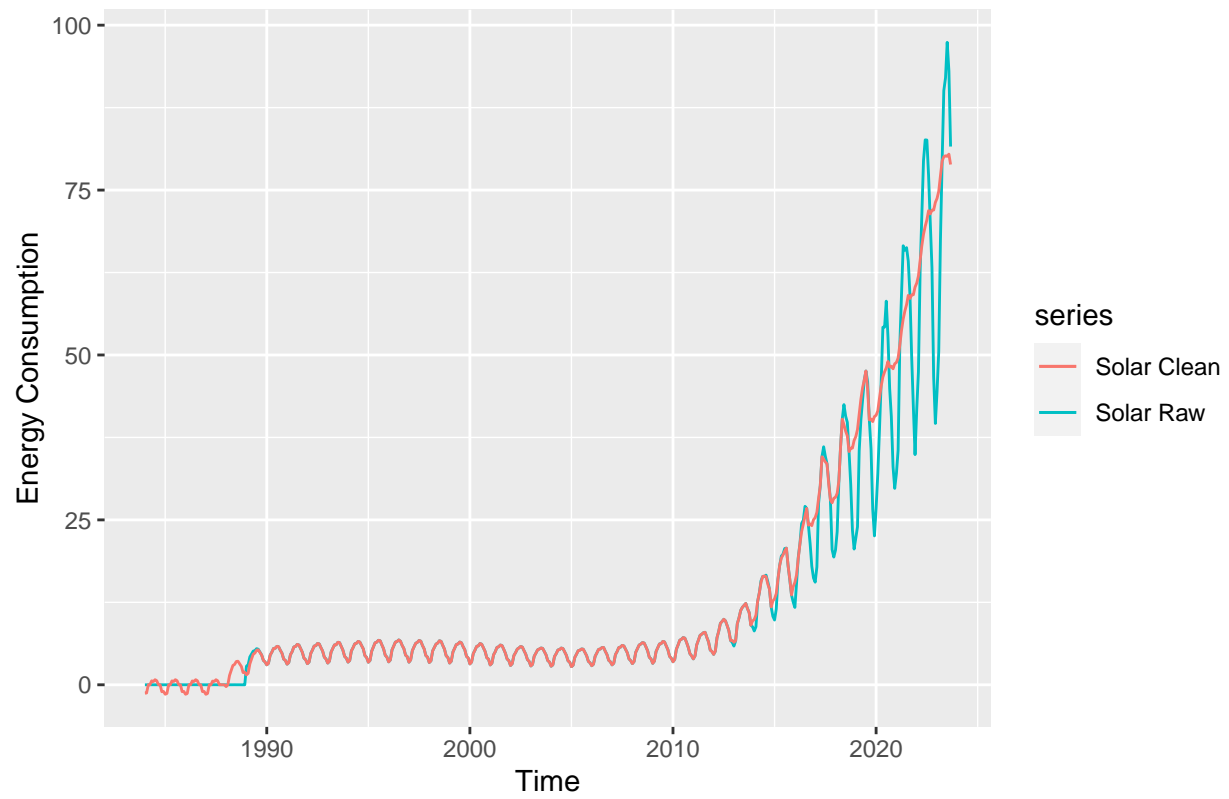
Apply the `tsclean()` to both series from Q7. Did the function removed any outliers from the series? Hint: Use `autoplot()` to check if there is difference between cleaned series and original series.

Answer: When considering substantial historical data in the data set, a ton of outliers are removed for solar and wind (you know this because the two trends are not super imposed). There seem to be a variety of factors (like new technology and increased environmental awareness) that have changed the energy consumption in these two categories. The variability across time tells us that historic data may not be the best predictor of current and future uses.

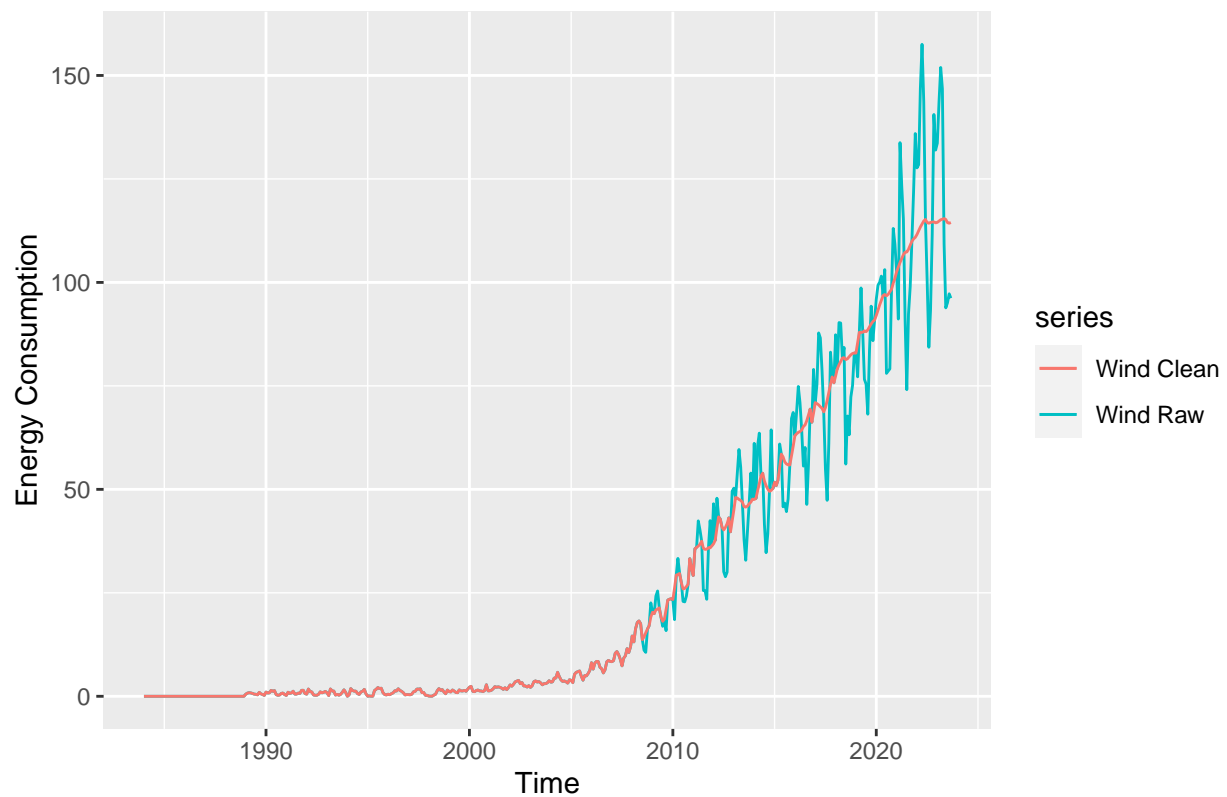
```
CleanWindTS <- tsclean(WindTS)

CleanSolarTS <- tsclean(SolarTS)

autoplot(SolarTS, series="Solar Raw") +
  autolayer(CleanSolarTS, series="Solar Clean") +
  ylab("Energy Consumption")
```



```
autoplot(WindTS, series="Wind Raw") +  
  autolayer(CleanWindTS, series="Wind Clean") +  
  ylab("Energy Consumption")
```



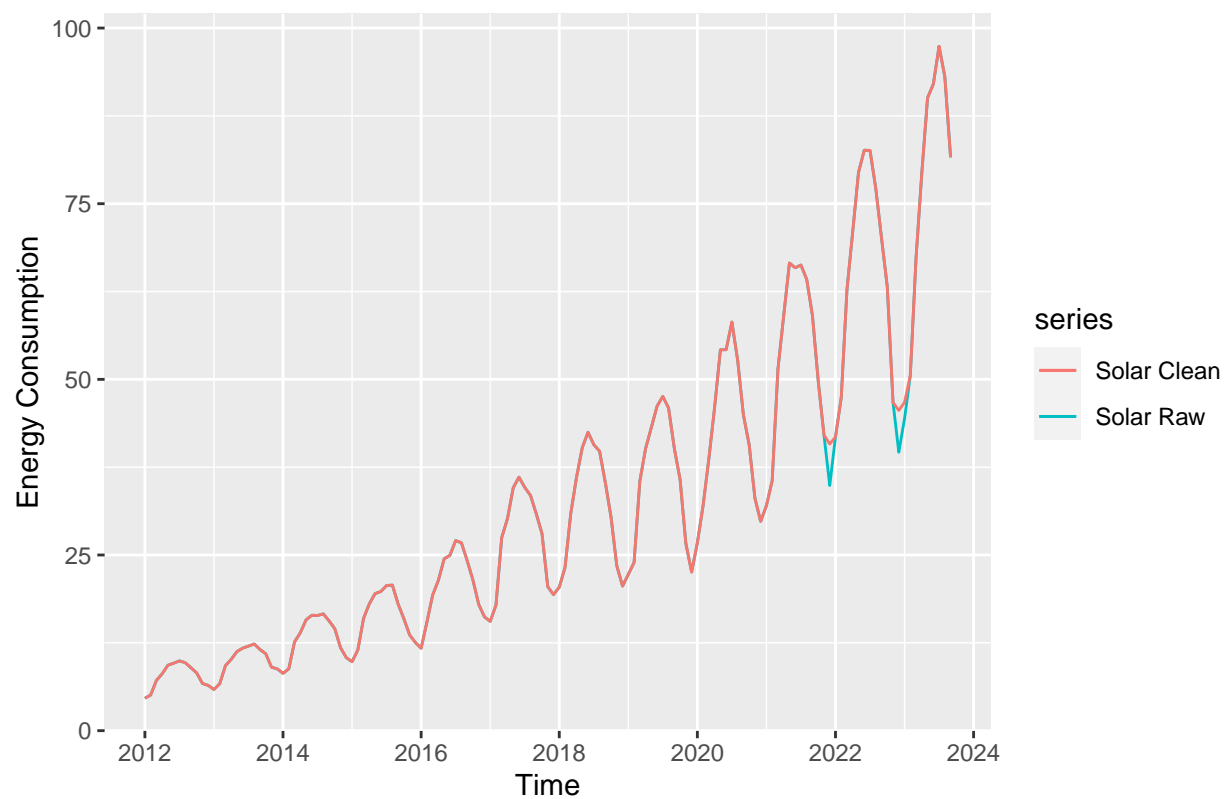
### Q9

Redo number Q8 but now with the time series you created on Q7, i.e., the series starting in 2014. Using what `autoplot()` again what happened now? Did the function removed any outliers from the series? After 2012

Answer: With the series including data only from 2012 on, there were very few outliers removed in the solar data (right around January of 2022 and 2023) but otherwise no data was removed (the two series are almost completely superimposed). This seems to show that it is better/ more predicatble to look at data around the same time rather than using largely historic data.

```
TSCWind <- tsclean(TSFilterWind)
TSCSolar <- tsclean(TSFilterSolar)

autoplot(TSFilterSolar, series="Solar Raw") +
  autolayer(TSCSolar, series="Solar Clean") +
  ylab("Energy Consumption")
```



```
autoplot(TSFilterWind, series="Wind Raw") +  
  autolayer(TSCWind, series="Wind Clean") +  
  ylab("Energy Consumption")
```

