Assignment 8: Time Series Analysis

Sydney Williams

Spring 2024

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on generalized linear models.

Directions

1. Rename this file `<FirstLast>_A08_TimeSeries.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

```r
# checking for working directory
getwd()
```

```
## [1] "/home/guest/EDA_Spring2024"
```

```r
# loading packages for assignment
library(tidyverse); library(lubridate); library(zoo); library(trend); library(readr); library(dplyr); li
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- 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
##
## Attaching package: 'zoo'
##
##
## The following objects are masked from 'package:base':
##
```

```
##     as.Date, as.Date.numeric
##
##
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
##
## here() starts at /home/guest/EDA_Spring2024
```

```
# setting ggplot theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
```

2. Import the ten datasets from the Ozone_TimeSeries folder in the Raw data folder. These contain ozone concentrations at Garinger High School in North Carolina from 2010-2019 (the EPA air database only allows downloads for one year at a time). Import these either individually or in bulk and then combine them into a single dataframe named `GaringerOzone` of 3589 observation and 20 variables.

```
#1

# Importing Ozone_Timeseries raw data files individually
EPAair_O3_GaringerNC2010_raw <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.cs
EPAair_O3_GaringerNC2011_raw <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.cs
EPAair_O3_GaringerNC2012_raw <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.cs
EPAair_O3_GaringerNC2013_raw <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.cs
EPAair_O3_GaringerNC2014_raw <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.cs
EPAair_O3_GaringerNC2015_raw <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.cs
EPAair_O3_GaringerNC2016_raw <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.cs
EPAair_O3_GaringerNC2017_raw <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.cs
EPAair_O3_GaringerNC2018_raw <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.cs
EPAair_O3_GaringerNC2019_raw <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.cs
EPAair_O3_GaringerNC2019_raw <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.cs

# Combining all raw data into one data frame using rbind
 GaringerOzone <- rbind(EPAair_O3_GaringerNC2010_raw,
                       EPAair_O3_GaringerNC2011_raw,
                       EPAair_O3_GaringerNC2012_raw,
                       EPAair_O3_GaringerNC2013_raw,
                       EPAair_O3_GaringerNC2014_raw,
                       EPAair_O3_GaringerNC2015_raw,
                       EPAair_O3_GaringerNC2016_raw,
                       EPAair_O3_GaringerNC2017_raw,
                       EPAair_O3_GaringerNC2018_raw,
                       EPAair_O3_GaringerNC2019_raw)
```

Wrangle

3. Set your date column as a date class.

4. Wrangle your dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE.

5. Notice there are a few days in each year that are missing ozone concentrations. We want to generate a daily dataset, so we will need to fill in any missing days with NA. Create a new data frame that

contains a sequence of dates from 2010-01-01 to 2019-12-31 (hint: `as.data.frame(seq())`). Call this new data frame Days. Rename the column name in Days to "Date".

6. Use a `left_join` to combine the data frames. Specify the correct order of data frames within this function so that the final dimensions are 3652 rows and 3 columns. Call your combined datla frame GaringerOzone.

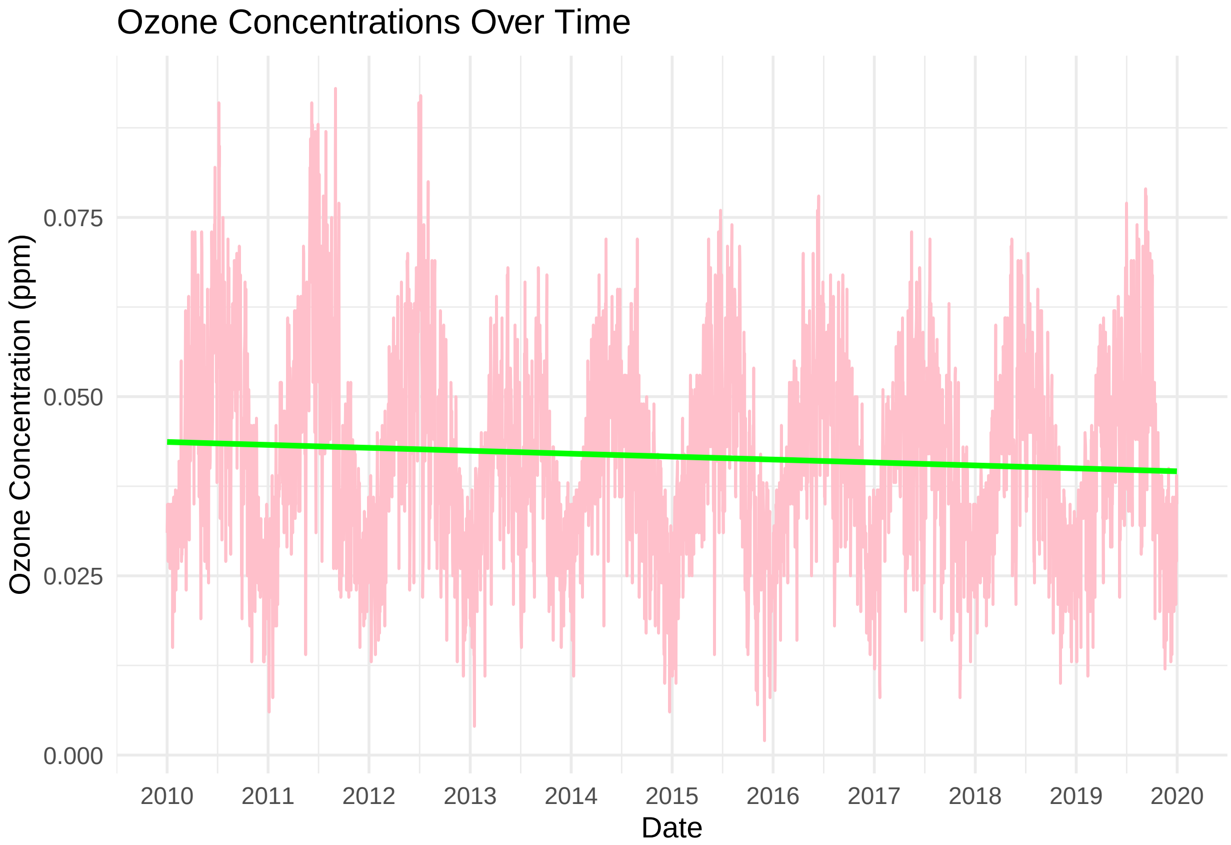
```
# 3
# setting date columns to date class
GaringerOzone$Date <- mdy(GaringerOzone$Date)

# 4
# Wrangling your dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentratio
GaringerOzone_filtered <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5
# Create a new data frame that contains a sequence of dates from 2010-01-01 to 2019-12-31
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "day"))
colnames(Days) <- "Date"

# 6
# Using a `left_join` to combine the data frames.
GaringerOzone <- left_join(Days, GaringerOzone_filtered, by = "Date")
```

Visualize

7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7
#  Create a line plot depicting ozone concentrations over time
GaringerOzone_lineplot <-
  ggplot(GaringerOzone_filtered, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line(color = "pink") +
  geom_smooth(method = "lm", se = FALSE, color = "green") +
  labs(x = "Date", y = "Ozone Concentration (ppm)", title = "Ozone Concentrations Over Time") +
  theme_minimal() +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year")



GaringerOzone_lineplot
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Answer: The line plot suggest that Ozone Concentration (ppm) keeps a steady trend over the years, decreasing only marginally through 2010-2020.

Time Series Analysis

Study question: Have ozone concentrations changed over the 2010s at this station?

8. Use a linear interpolation to fill in missing daily data for ozone concentration. Why didn't we use a piecewise constant or spline interpolation?

```
#8
# Using a linear interpolation to fill in missing daily data for ozone concentration
GaringerOzone <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentrati
```

Answer: Piecewise constant interpolation would only fill in the missing values with the value of the nearest known data point. Because of this, it might not accurately convey the true trend of data known between known points. Spline interpolation is not needed here because the data is not as complex for quadratic data.

9. Create a new data frame called `GaringerOzone.monthly` that contains aggregated data: mean ozone concentrations for each month. In your pipe, you will need to first add columns for year and month to form the groupings. In a separate line of code, create a new Date column with each month-year combination being set as the first day of the month (this is for graphing purposes only)

```
#9

# Creating a new data frame called 'GaringerOzone.monthly`
```

```
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(Year = lubridate::year(Date),
         Month = lubridate::month(Date))  %>%
  mutate(Date = ymd(paste(Year, Month, "01", sep = "-"))) %>%
  group_by(Year, Month, Date) %>%
  summarize(mean_ozone = mean(Daily.Max.8.hour.Ozone.Concentration.clean, na.rm = TRUE)) %>%
  select(Date,mean_ozone)
```

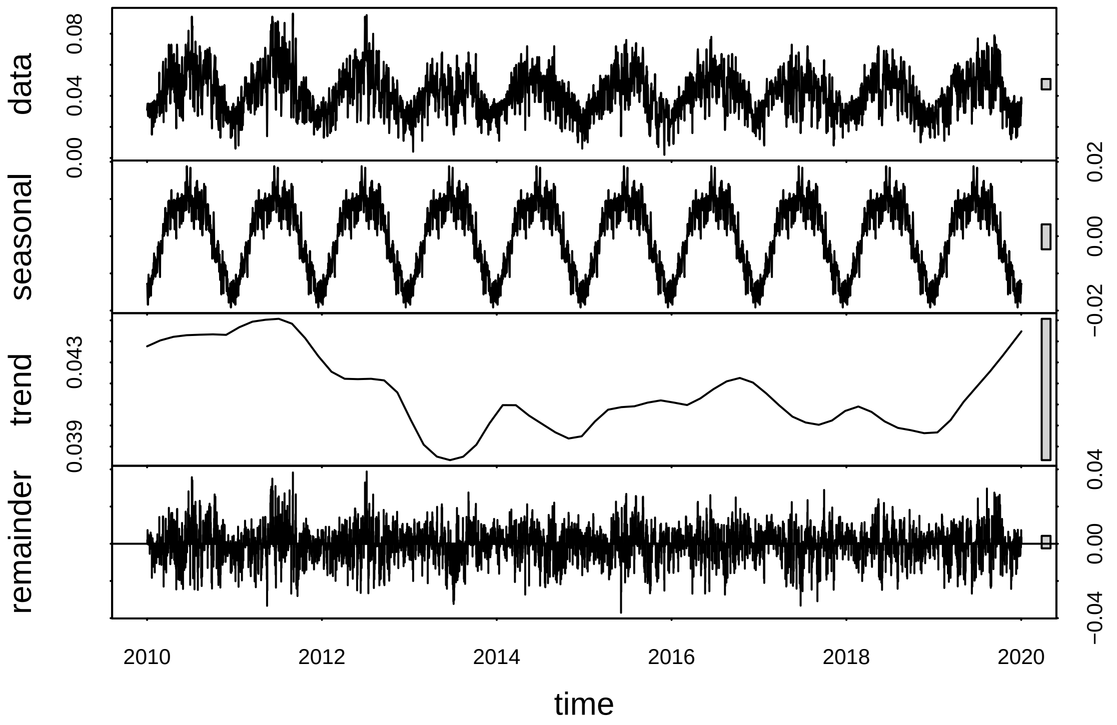
```
## `summarise()` has grouped output by 'Year', 'Month'. You can override using the
## `.groups` argument.
## Adding missing grouping variables: `Year`, `Month`
```

10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

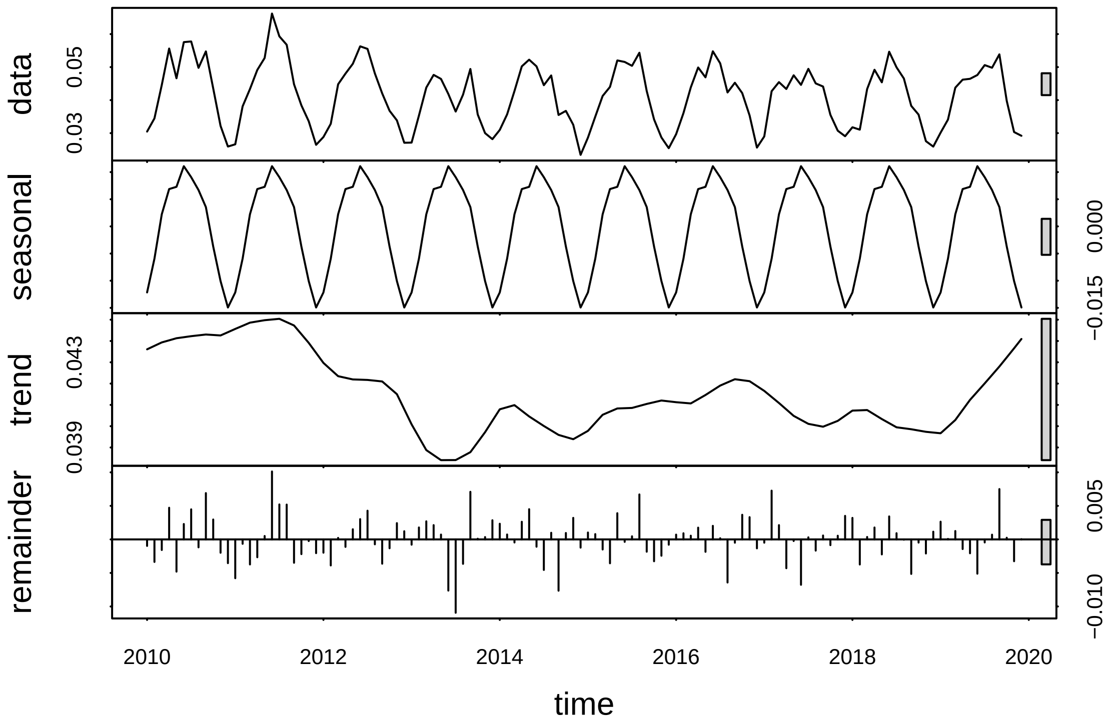
```
#10
# Creating time series object for daily observations
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration.clean,
                          start = c(year(first(GaringerOzone$Date)),month(first(GaringerOzone$Date))),
                             frequency = 365)

# Creating time series object for monthly average ozone values
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone,
                               start = c(year(first(GaringerOzone.monthly$Date)),
                                         month(first(GaringerOzone.monthly$Date))),
                               frequency = 12)
```

11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11

# Decomposing ozone daily time series
ozonedaily_decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")

# Decomposing ozone monthly time series
ozonemonthly_decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")

# Plotting the decomposed components of the daily time series
plot(ozonedaily_decomposed)
```



```
# Plotting the decomposed components of the monthly time series
plot(ozonemonthly_decomposed)
```



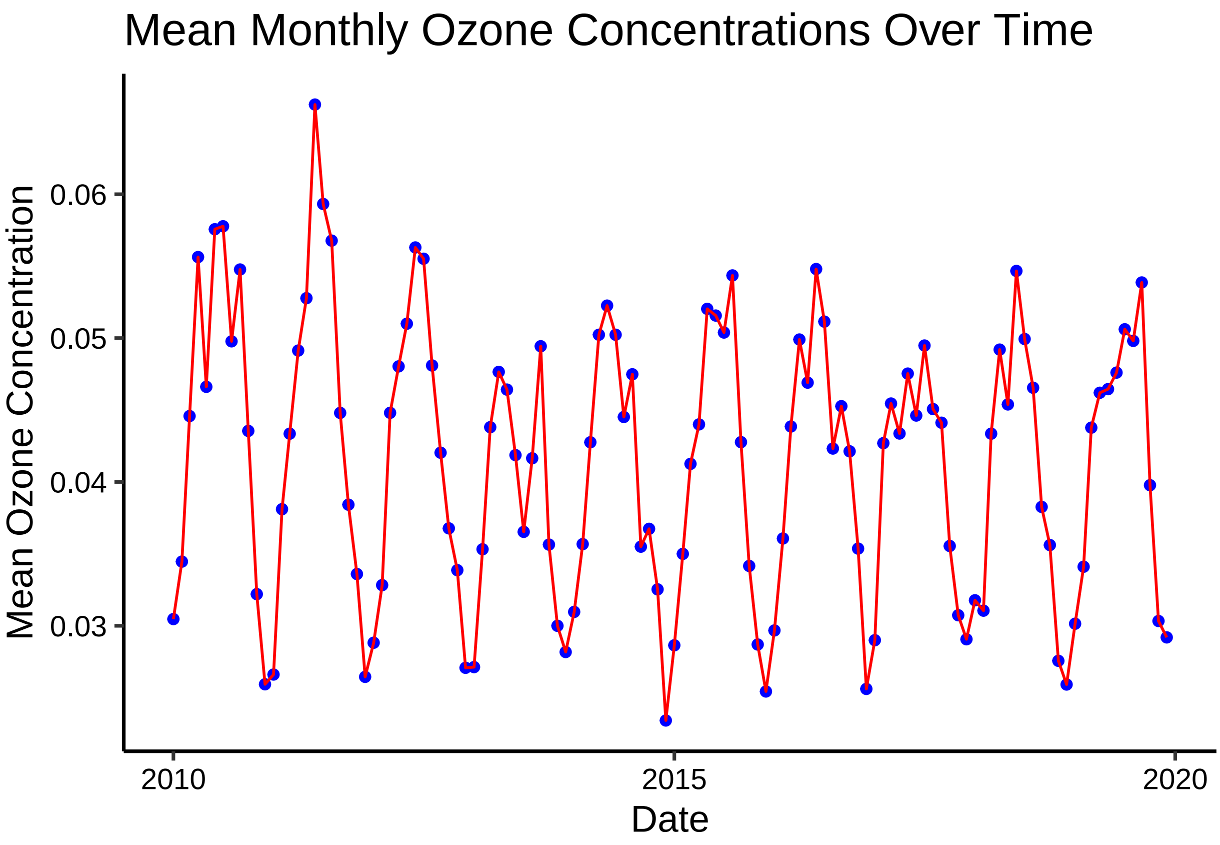
12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

```
#12

monthly_Ozone<- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
```

Answer: The seasonal Mann Kendall test is most appropriate here because the ozone concentration data is based on seasonality where there is fluctuations in data. Because we don't want this to influence our data, it is removed to ensure that the trend analysis accurately reflects the underlying trends in the data while accounting for seasonal variations. Using the regular Mann Kendall test wouldn't be as accurate because we could easily point to seasonality versus other underlying trends as to why the ozone concentrations are fluctuating.

13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. Edit your axis labels accordingly.

```
# 13
# Creating a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a g
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_ozone)) +
  geom_point(color = "blue") +
  geom_line(color = "red") +
  labs(x = "Date", y = "Mean Ozone Concentration", title = "Mean Monthly Ozone Concentrations Over Time
```



14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

    Answer: The plot illustrates the mean monthly ozone concentrations over time, revealing a general trend of fluctuation with periods of higher and lower concentrations. This suggests the presence of variability in ozone levels throughout the observation period. The statistical analysis using the Mann-Kendall test indicates a p-value of 0.05 significant trend in ozone concentrations over time, which indicates that there is a strong relationship between mean ozone concentration over time.

15. Subtract the seasonal component from the `GaringerOzone.monthly.ts`. Hint: Look at how we extracted the series components for the EnoDischarge on the lesson Rmd file.

16. Run the Mann Kendall test on the non-seasonal Ozone monthly series. Compare the results with the ones obtained with the Seasonal Mann Kendall on the complete series.

```
#15
# Extract the seasonal component
ozonemonthly_nonseasonalcomponents <- as.data.frame(ozonemonthly_decomposed$time.series[,1:3])

ozonemonthly_nonseasonalcomponents  <-
  mutate(ozonemonthly_nonseasonalcomponents,
        Observed = GaringerOzone.monthly$mean_ozone,
        Date = GaringerOzone.monthly$Date)

ozonemonthly_nonseasonalcomponents_ts <- ts(ozonemonthly_nonseasonalcomponents[,1:3], frequency = 12)
```

```r
#16

# Running Mann-Kendall trend test on the non-seasonal ozone monthly series
non_seasonal_ozone <- MannKendall(ozonemonthly_nonseasonalcomponents_ts)
```

Answer: The p-value for the seasonal Mann Kendall test is less than 0.05 which indicates a strong trend, versus no trend. The p-value for the Mann Kendall test is higher than 0.05 indicating a weaker trend.