Assignment 8: Time Series Analysis

Becca Cox

Fall 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
getwd()
```

```
## [1] "/home/guest/EDE_Fall2024/EDE_Fall2024"
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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
```

```
library(lubridate)
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(trend)

my_theme <- theme(panel.background =
          element_rect(fill = 'lavenderblush1'),
          plot.background = element_rect(fill = "lavenderblush2"),
          axis.text = element_text(color = "darkslateblue"),
          axis.title = element_text(color = "darkslateblue"),
          plot.title = element_text(color = "darkslateblue"),
          legend.title = element_text(color = "darkslateblue"),
          legend.key = element_rect(fill = 'lavenderblush1'),
          legend.background = element_rect(fill = 'lavenderblush2'),
          legend.position = 'top')

theme_set(my_theme)
```

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

epaozone_2010 <- read.csv(("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
                          stringsAsFactors = TRUE)

epaozone_2011 <- read.csv(("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
                          stringsAsFactors = TRUE)

epaozone_2012 <- read.csv(("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
                          stringsAsFactors = TRUE)

epaozone_2013 <- read.csv(("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
                          stringsAsFactors = TRUE)

epaozone_2014 <- read.csv(("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
                          stringsAsFactors = TRUE)

epaozone_2015 <- read.csv(("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
                          stringsAsFactors = TRUE)

epaozone_2016 <- read.csv(("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
                          stringsAsFactors = TRUE)
```

```
epaozone_2017 <- read.csv(("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
                          stringsAsFactors = TRUE)

epaozone_2018 <- read.csv(("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
                          stringsAsFactors = TRUE)

epaozone_2019 <- read.csv(("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
                          stringsAsFactors = TRUE)

GaringerOzone <- bind_rows(epaozone_2010,epaozone_2011,epaozone_2012,epaozone_2013,
                      epaozone_2014,epaozone_2015,epaozone_2016,epaozone_2017,epaozone_2018,
                      epaozone_2019)
```

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 data frame GaringerOzone.

```
#3

GaringerOzone <- GaringerOzone %>%
  mutate(Date = mdy(Date))

# 4

GaringerOzone_processed <- GaringerOzone %>%
  select(Date,Daily.Max.8.hour.Ozone.Concentration,DAILY_AQI_VALUE)

# 5

Days <- as.data.frame(seq(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"), by = "day"))

colnames(Days) <- c("Date")

# 6

GaringerOzone_joined <- left_join(Days, GaringerOzone_processed, by = "Date")
```

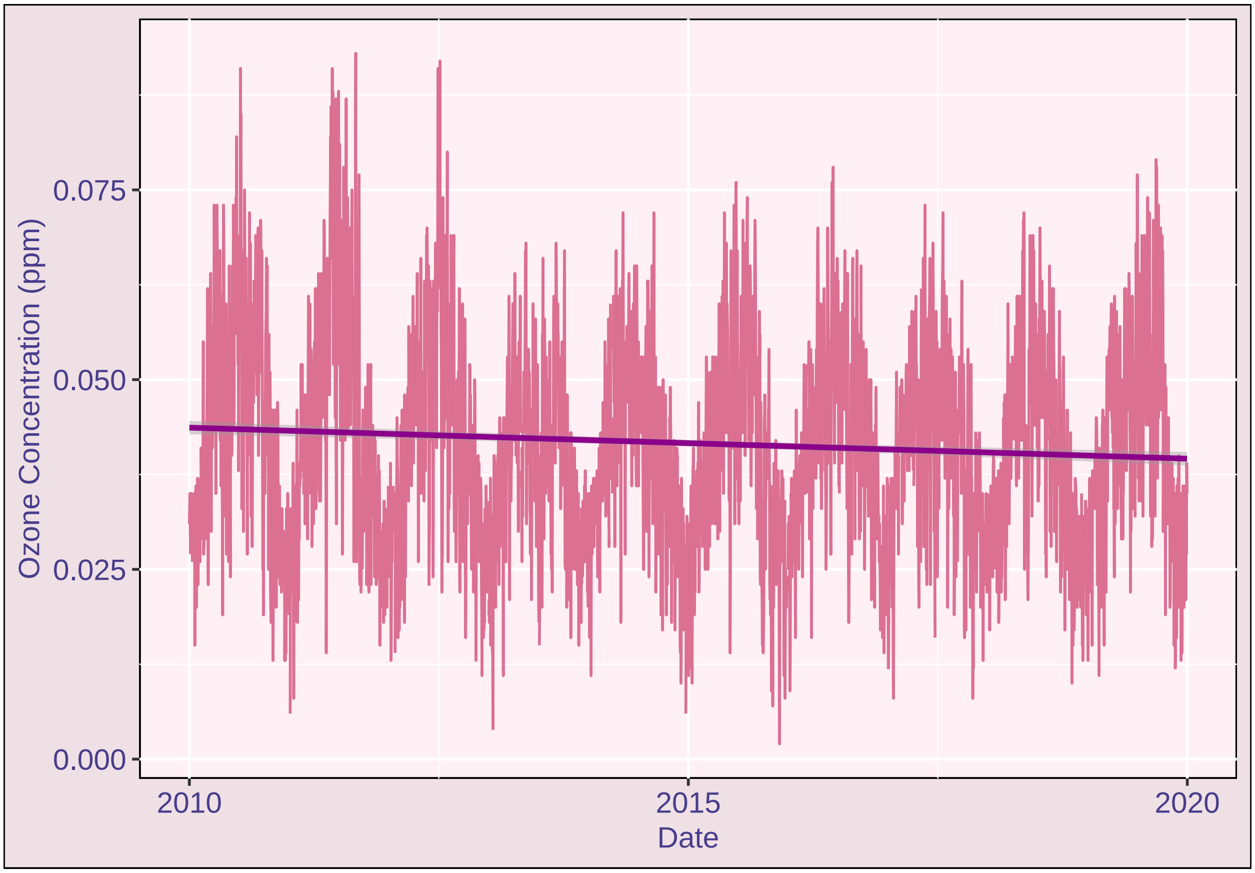
Visualize

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

ozoneovertime <- ggplot(GaringerOzone_joined, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line(colour = "palevioletred")+
  geom_smooth(method = "lm", color = "darkmagenta") +
  ylab("Ozone Concentration (ppm)")

print(ozoneovertime)
```

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

## Warning: Removed 63 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```



Answer: The line plot suggests that there is a downward trend in ozone concentration over time.

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

summary(GaringerOzone_joined$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

```
ozone_data_clean <-
  GaringerOzone_joined %>%
  mutate(ozone.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration)) %>%
  select(Date,ozone.clean)

summary(ozone_data_clean$ozone.clean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## 0.00200 0.03200 0.04100 0.04151 0.05100 0.09300
```

```
#NA is gone
```

> Answer: The plot above suggests that the data have a fairly slow, gradual slope downward, so they likely do not require a curved, quadratic function to interpolate points where data are missing. Both piecewise constant and spline methods of interpolation involve quadratic equations. A straight-line approach to interpolation is effective in interpolating the data in a dataset with this trend.

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)

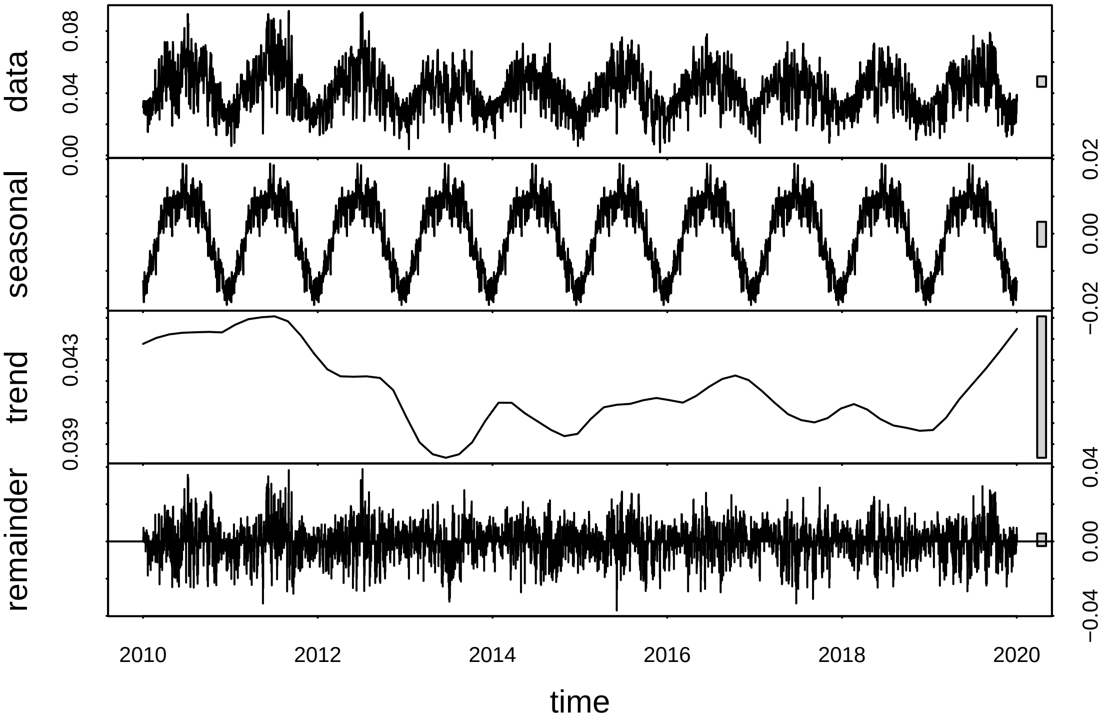
```
#9

library(dplyr)

GaringerOzone.monthly <- ozone_data_clean %>%
  separate(Date,c("Year","Month","Day")) %>%
  mutate(Date = paste0(Month,"-","01","-",Year)) %>%
  group_by(Month,Year) %>%
  mutate(monthlymean = mean(ozone.clean)) %>%
 distinct(Month,Year, .keep_all = TRUE)%>%
  select(Month,Year,Date,monthlymean)


GaringerOzone.monthly$Date <- mdy(GaringerOzone.monthly$Date)
```

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.

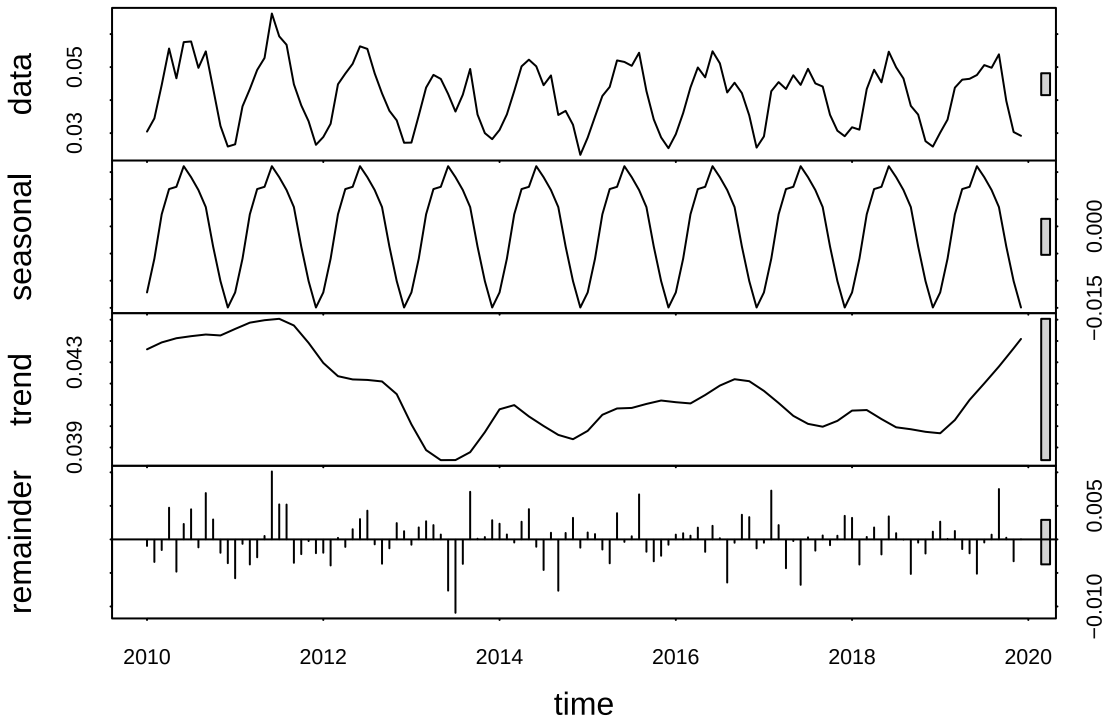
```
#10

GaringerOzone.daily.ts <- ts(ozone_data_clean$ozone.clean, start = c(2010,1), frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$monthlymean, start = c(2010,1), frequency = 12)
```

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

```
#11

ozone.daily.decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")

plot(ozone.daily.decomposed)
```



```
ozone.monthly.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")

plot(ozone.monthly.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

library(Kendall)

montly.ozone.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

montly.ozone.trend
```

```
## tau = -0.143, 2-sided pvalue =0.046724
```

```
summary(montly.ozone.trend)
```

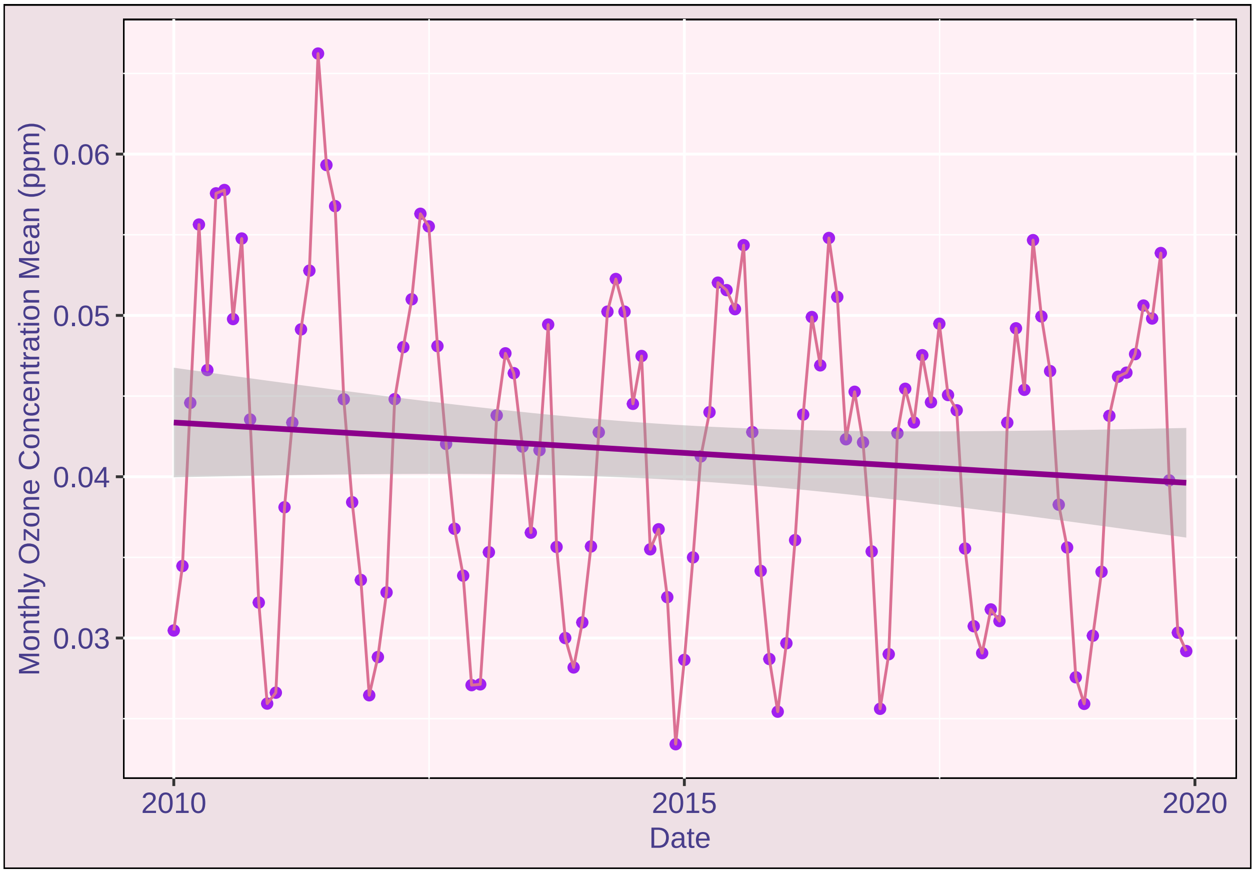
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The decomposition of the time series data indicates that there is seasonality to this dataset. The data set is also made up of asymmetrical continuous data, so is considered non-parametric. These characteristics best match the requirements of a seasonal Mann-Kendall test.

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.

```r
# 13

monthly.ozone.over.time <- ggplot(GaringerOzone.monthly, aes(x = Date, y = monthlymean)) +
  geom_point(color = "purple")+
  geom_line(colour = "palevioletred")+
  geom_smooth(method = "lm", color = "darkmagenta")+
  ylab("Monthly Ozone Concentration Mean (ppm)")

print(monthly.ozone.over.time)
```

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



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: Ozone concentrations have decreased at this station over the 2010s. The null hypothesis is that there is no monotonic trend; the pvalue of 0.0467, which is less that the significance level of 0.05, means that we can reject this null hypothesis. The tau value of -0.143 indicates that the trend is decreasing 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

ozone.timeseries.components <- as.data.frame(ozone.monthly.decomposed$time.series[,1:3])

ozone.timeseries.components <- mutate(ozone.timeseries.components,
        Observed = GaringerOzone.monthly$monthlymean,
        Date = GaringerOzone.monthly$Date)

ozone.timeseries.components <- mutate(ozone.timeseries.components,
        NoSeason = (ozone.timeseries.components$Observed - ozone.timeseries.components$seasonal))

#16

ozone.noseason.ts <- ts(ozone.timeseries.components$NoSeason, start = c(2010,1), frequency = 12)

noseason.trend <- Kendall::MannKendall(ozone.noseason.ts)
noseason.trend
```

```
## tau = -0.165, 2-sided pvalue =0.0075402
```

```
summary(noseason.trend)
```

```
## Score =  -1179 , Var(Score) = 194365.7
## denominator =  7139.5
## tau = -0.165, 2-sided pvalue =0.0075402
```

```
montly.ozone.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
montly.ozone.trend
```

```
## tau = -0.143, 2-sided pvalue =0.046724
```

```
summary(montly.ozone.trend)
```

```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The non-seasonal Mann-Kendall test provided a p-value of 0.0075 and a tau value of -0.165. These are similar to the results of the seasonal Mann-Kendall test: there is a monotonic trend, and it is decreasing over time. The p-values are different between the non-seasonal and seasonal tests, likely because we completely removed seasonality from the dataset for the non-seasonal test, and the relationship between the residual values may differ from the relationships between values when seasonality is taken into account. Despite these differences, both tests provided p-values below the level of significance, so we can reject the null hypothesis in both cases.