Assignment 8: Time Series Analysis

Alex Lopez

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

```
#load packages
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 error
```

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

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

```
library(trend)
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
```

```
#check wd
here()
```

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

```
#set theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        axis.title.x = element_text(size = 10),
        axis.title.y = element_text(size = 10),
        axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "right",
        legend.title = element_text(size = 10),
        legend.text = element_text(size = 8),
        plot.title = element_text(hjust = 0.5, size = 12))
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

#list all datasets
list.datasets <- list.files("Data/Raw/Ozone_TimeSeries/", pattern = "*.csv",
                            full.names = TRUE)

#read and combine files
GaringerOzone <- lapply(list.datasets, read.csv, stringsAsFactors = TRUE) %>%
  bind_rows()
```

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

#format date
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = '%m/%d/%Y')

# 4

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

# 5

Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"),
                          by = 'day'))
#rename
names(Days) <- 'Date'

# 6

GaringerOzone <- Days %>%
  left_join(GaringerOzone, by = "Date")

#check that there are 3652 rows and 3 columns
dim(GaringerOzone)
```

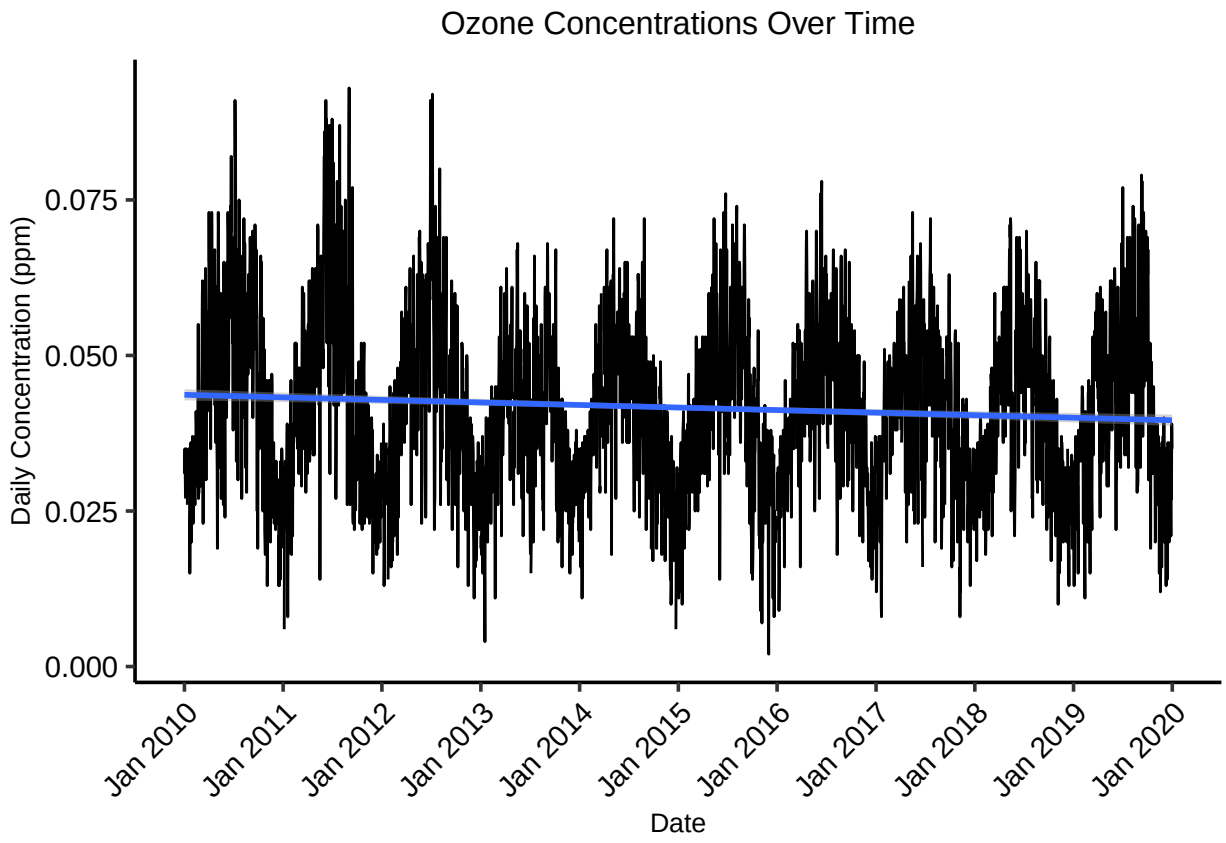
```
## [1] 3652    3
```

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

ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = 'lm') +
  scale_x_date(date_breaks = '1 year', date_labels = '%b %Y') +
  labs(
    title = "Ozone Concentrations Over Time",
    x = "Date",
    y = "Daily Concentration (ppm)",
  )
```

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

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



Answer: According to the plot, there appears to be a very slight decrease 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

GaringerOzone.Clean <- GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

#compare NAs
summary(GaringerOzone$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
```

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

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

Answer: We didn't use a piecewise constant interpolation, because the missing data would equal the nearest measurement to that date, which might result in sudden jumps between the values, which might not represent the actual trend, and it also wouldn't reflect gradual daily changes. We didn't use spline interpolation, because the data we're working with doesn't appear to be continuous or data that would have a curved pattern, especially with missing days.

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

GaringerOzone.monthly <- GaringerOzone.Clean %>%
  mutate(
    Month = month(Date), Year = year(Date)
    ) %>%
  group_by(Year, Month) %>%
  summarise(MonthlyMeanOzone = mean(Daily.Max.8.hour.Ozone.Concentration))
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

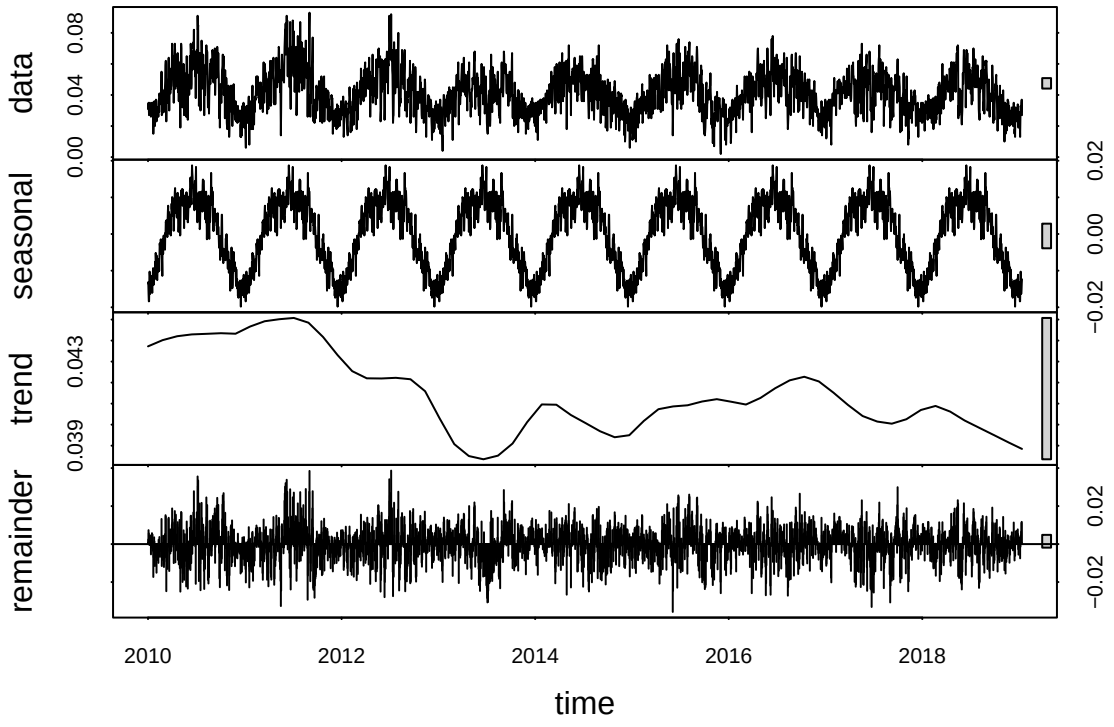
```
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = paste(Year, Month, "01", sep = "-"))
```

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.

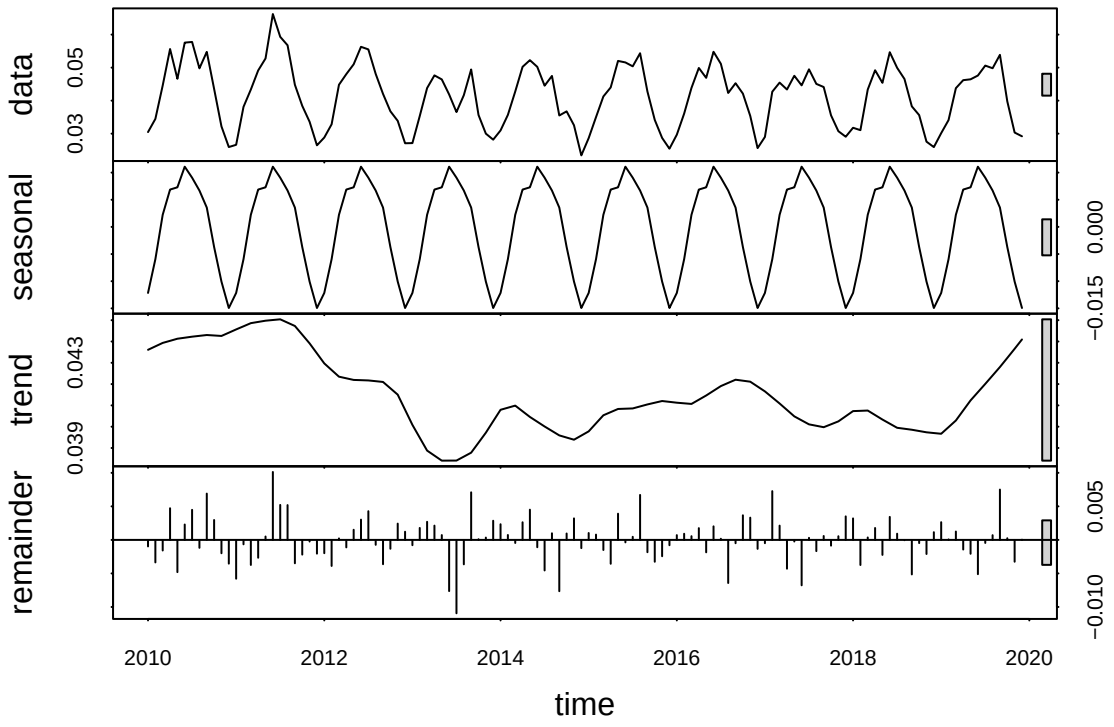
```
#10

GaringerOzone.daily.ts <-
  ts(GaringerOzone.Clean$Daily.Max.8.hour.Ozone.Concentration,
     start = c(2010,1), end = c(2019,12), frequency = 365)

GaringerOzone.monthly.ts <-
  ts(GaringerOzone.monthly$MonthlyMeanOzone, start = c(2010,1),
     end = c(2019,12), frequency = 12)
```

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

```
#11

GaringerOzone.Clean.Decomposed <-
  stl(GaringerOzone.daily.ts, s.window = 'periodic')

plot(GaringerOzone.Clean.Decomposed)
```




```
GaringerOzone.monthly.Decomposed <-
  stl(GaringerOzone.monthly.ts, s.window = 'periodic')

plot(GaringerOzone.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

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

GaringerOzone.monthly.trend
```

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

```
summary(GaringerOzone.monthly.trend)
```

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

```
GaringerOzone.monthly.trend2 <- trend::smk.test(GaringerOzone.monthly.ts)

GaringerOzone.monthly.trend2
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
```

```
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

```
summary(GaringerOzone.monthly.trend2)
```

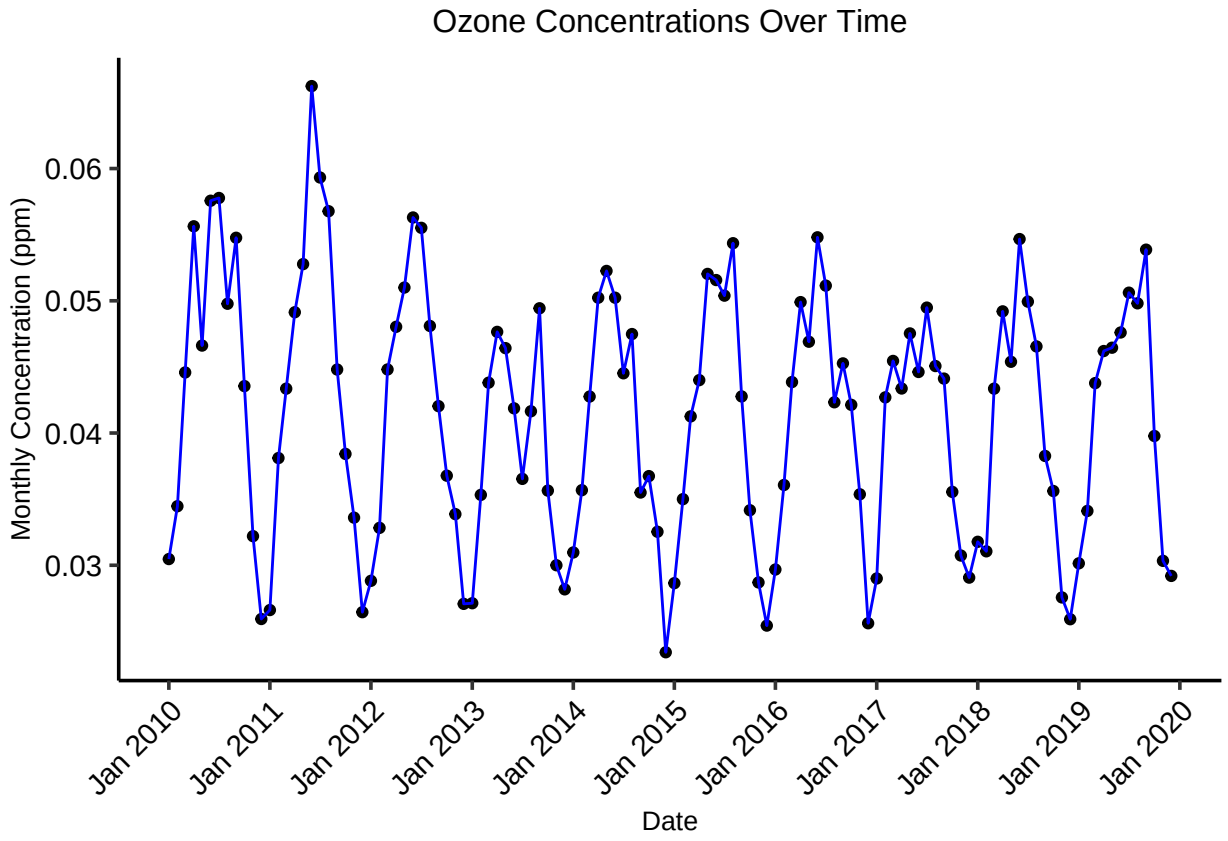
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                       S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 2:   S = 0   -1  125 -0.022  0.000  1.00000
## Season 3:   S = 0   -4  124 -0.090 -0.269  0.78762
## Season 4:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 5:   S = 0  -15  125 -0.333 -1.252  0.21050
## Season 6:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 7:   S = 0  -11  125 -0.244 -0.894  0.37109
## Season 8:   S = 0   -7  125 -0.156 -0.537  0.59151
## Season 9:   S = 0   -5  125 -0.111 -0.358  0.72051
## Season 10:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 11:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 12:   S = 0  11  125  0.244  0.894  0.37109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: Since the seasonal Mann-Kendall test is suited for monotonic trend analysis in a time series that exhibits seasonality and is non-parametric, the seasonal Mann-Kendall is most appropriate in this case. The formation of ozone depends on the amount of sunlight and temperature, which, naturally, makes monthly ozone concentration seasonal due to the changes in sunlight and temperature according to season.

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

GaringerOzone.monthly$Date <- as.Date(GaringerOzone.monthly$Date)

ggplot(GaringerOzone.monthly, aes(x = Date, y = MonthlyMeanOzone)) +
  geom_point() +
  geom_line(color = 'blue') +
  scale_x_date(date_breaks = '1 year', date_labels = '%b %Y') +
```

```
labs(
  title = "Ozone Concentrations Over Time",
  x = "Date",
  y = "Monthly Concentration (ppm)",
)
```



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: There appears to be a seasonal pattern, with concentrations peaking during the summer months and dropping in the winter months, consistent with the expected seasonal variability in ozone concentration. The Seasonal Mann-Kendall test results indicate that there is a significant monotonic downward trend in monthly ozone concentrations in the 2010s (τ = -0.143, p $<$ 0.05). We therefore reject the null hypothesis that there is no trend in ozone levels over the observed period.

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.

```r
#15

GaringerOzone.monthly.woSeasonal <-
  GaringerOzone.monthly.ts - GaringerOzone.monthly.Decomposed$time.series[, "seasonal"]

GaringerOzone.monthly.woSeasonal.ts <- ts(GaringerOzone.monthly.woSeasonal,
                                          start = start(GaringerOzone.monthly.ts),
                                          frequency = frequency(GaringerOzone.monthly.ts))

#16

GaringerOzone.monthly.woSeasonal.trend <- Kendall::MannKendall(GaringerOzone.monthly.woSeasonal.ts)

GaringerOzone.monthly.woSeasonal.trend
```

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

```r
summary(GaringerOzone.monthly.woSeasonal.trend)
```

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

Answer: In comparison with the results obtained with the Seasonal Mann Kendall test on the complete series, the p-value for the Mann Kendall test on the non-seasonal Ozone monthly series is also less than 0.05. However, the p-value is even smaller after removing the seasonal component. This could mean that the trend in ozone concentrations over the observed time period could be due to another factor other than seasonality.