Assignment 8: Time Series Analysis

Laura Exar

Spring 2023

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

```
#1
#Check working directory
library(here)
```

```
## here() starts at /home/guest/R/EDA-Spring2023
```

```
here()
```

```
## [1] "/home/guest/R/EDA-Spring2023"
```

```
#Loaded packages
library(tidyverse); library(lubridate); library(zoo); library(trend)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2
## --
```

```
## v ggplot2 3.4.0      v purrr   1.0.0
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts --------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
##
## Attaching package: 'lubridate'
##
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
##
##
##
## Attaching package: 'zoo'
##
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
#Set theme
library(ggthemes)

my_theme <- theme_base() +
  theme(axis.line = element_line(
      linewidth = 1,
      colour = "black"),
    plot.background = element_rect(
      color='grey'),
    plot.title = element_text(
      color='blue'),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 5))

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.

```
#2
#Imported the datasets into one dataframe
GaringerOzone <- list.files(path = "Data/Raw/Ozone_TimeSeries",
                            pattern = "*.csv", full.names = TRUE) %>%
  lapply(read_csv) %>%
  bind_rows
GaringerOzone
```

```
## # A tibble: 3,589 x 20
```

```
##    Date       Source Site ~1   POC Daily~2 UNITS DAILY~3 Site ~4 DAILY~5 PERCE~6
##    <chr>      <chr>    <dbl> <dbl>   <dbl> <chr>   <dbl> <chr>     <dbl>   <dbl>
##  1 01/01/2010 AQS     3.71e8     1   0.031 ppm        29 Garing~      17     100
##  2 01/02/2010 AQS     3.71e8     1   0.033 ppm        31 Garing~      17     100
##  3 01/03/2010 AQS     3.71e8     1   0.035 ppm        32 Garing~      17     100
##  4 01/04/2010 AQS     3.71e8     1   0.031 ppm        29 Garing~      17     100
##  5 01/05/2010 AQS     3.71e8     1   0.027 ppm        25 Garing~      17     100
##  6 01/07/2010 AQS     3.71e8     1   0.033 ppm        31 Garing~      17     100
##  7 01/08/2010 AQS     3.71e8     1   0.035 ppm        32 Garing~      17     100
##  8 01/09/2010 AQS     3.71e8     1   0.032 ppm        30 Garing~      17     100
##  9 01/10/2010 AQS     3.71e8     1   0.032 ppm        30 Garing~      17     100
## 10 01/11/2010 AQS     3.71e8     1   0.03  ppm        28 Garing~      17     100
## # ... with 3,579 more rows, 10 more variables: AQS_PARAMETER_CODE <dbl>,
## #   AQS_PARAMETER_DESC <chr>, CBSA_CODE <dbl>, CBSA_NAME <chr>,
## #   STATE_CODE <dbl>, STATE <chr>, COUNTY_CODE <dbl>, COUNTY <chr>,
## #   SITE_LATITUDE <dbl>, SITE_LONGITUDE <dbl>, and abbreviated variable names
## #   1: 'Site ID', 2: 'Daily Max 8-hour Ozone Concentration',
## #   3: DAILY_AQI_VALUE, 4: 'Site Name', 5: DAILY_OBS_COUNT, 6: PERCENT_COMPLETE
```

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
#Set date column as a date class
GaringerOzone$Date <- mdy(GaringerOzone$Date)

#4
#Wrangled dataset so that it only contains three columns
colnames(GaringerOzone)[colnames(GaringerOzone) == "Daily Max 8-hour Ozone Concentration"] = "Daily.Max

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

#5
#Filled in any missing days with NA and created a new data frame
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "1 day"))
colnames(Days) = c("Date") #Renamed the column name in Days to "Date"

#6
```

```
#Combined the data frames
GaringerOzone <- left_join(Days, GaringerOzone)
```

```
## Joining, by = "Date"
```

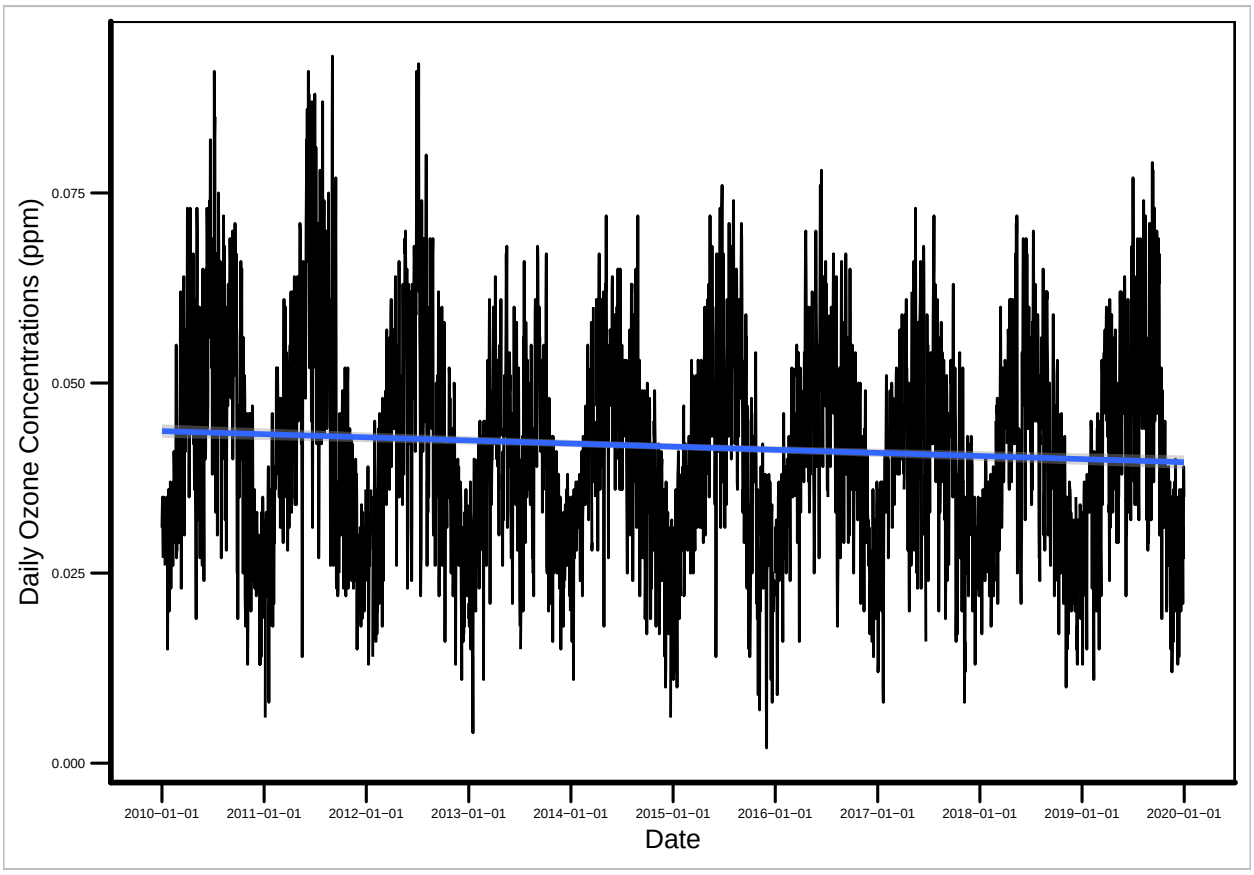
Visualize

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
#Created a line plot
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  scale_x_date(date_breaks="1 year") +
  labs(x = "Date", y = "Daily Ozone Concentrations (ppm)") +
  geom_smooth(method="lm")
```

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

```
## Warning: Removed 63 rows containing non-finite values (`stat_smooth()`).
```



Answer: Yes, the plot shows a slight decrease in ozone concentration over time. There also appears to be seasonality in the data.

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
#Filled in missing data using linear interpolation
GaringerOzone_Clean <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration.Clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentrati
```

Answer: We used a linear interpolation because we believe that our data has a linear trend, so it would make sense to use a linear interpolation to draw a straight line between the known points rather than a piecewise constant (which uses nearest neighbor) and a spline (which uses a quadratic function).

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)

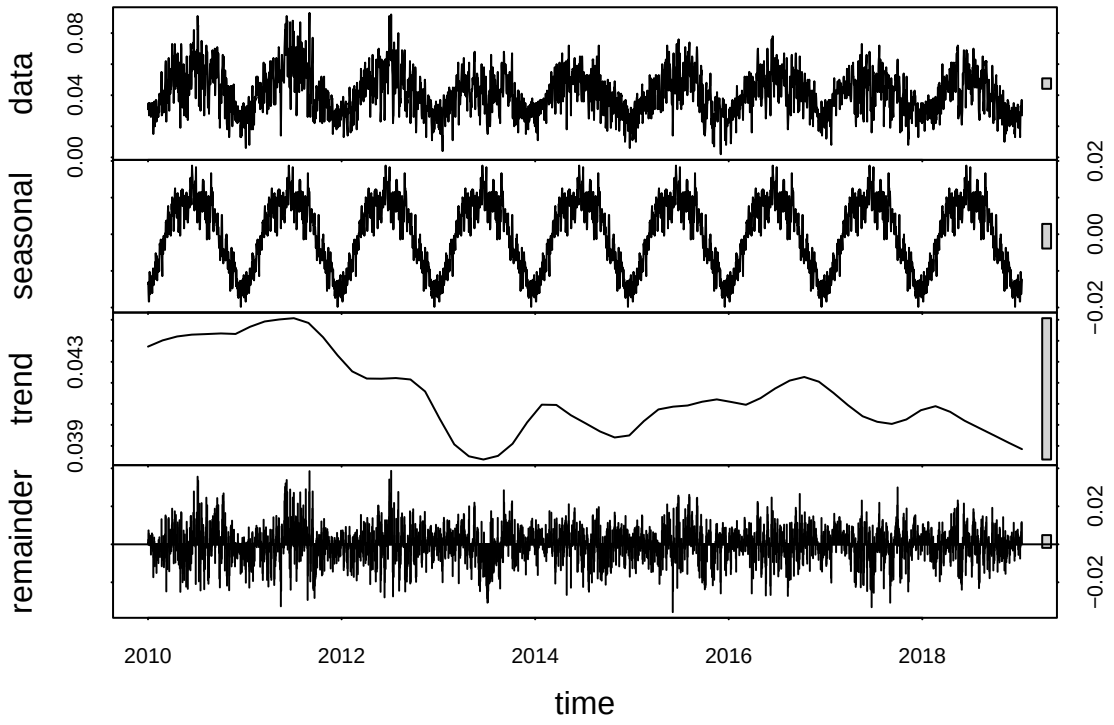
```
#9
#Created a new data frame
GaringerOzone.monthly <- GaringerOzone_Clean %>%
  mutate(Month = month(Date)) %>%
  mutate(Year = year(Date)) %>%
  mutate(Day = "01") %>%
  mutate(Date.MY = mdy(paste(Month,Day,Year, sep = "-"))) %>%
  group_by(Month) %>%
  mutate(Mean.Monthly.Ozone =
           mean(Daily.Max.8.hour.Ozone.Concentration.Clean))
```

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.

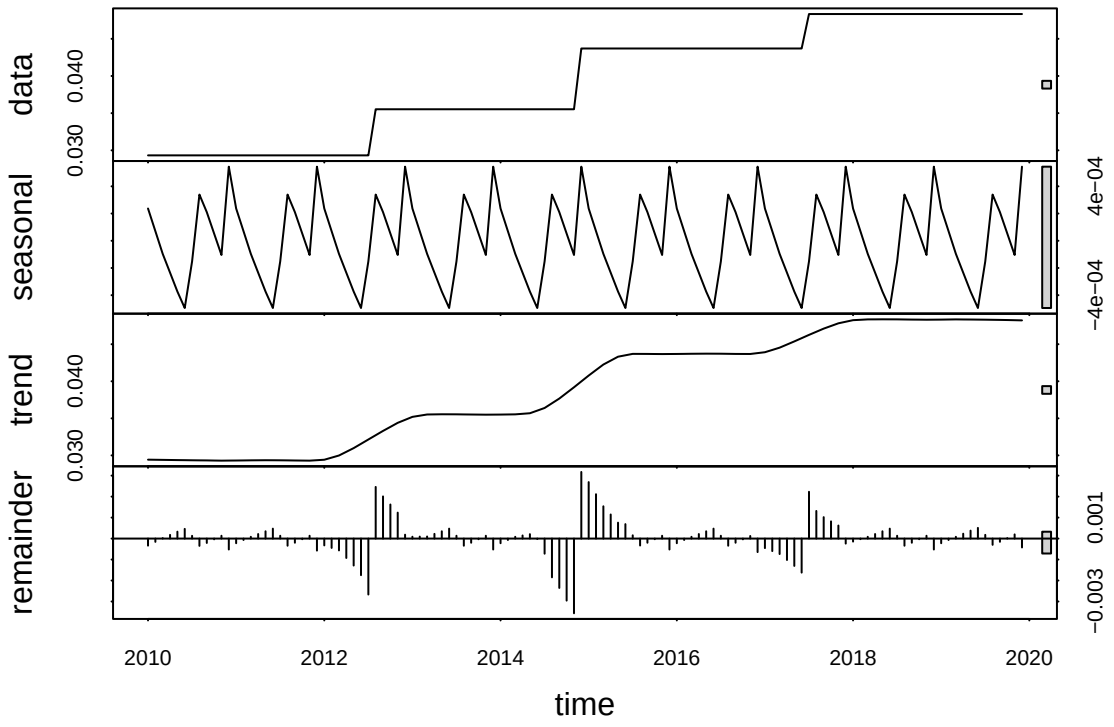
```
#10
#Created a daily time series
GaringerOzone.daily.ts <- ts(GaringerOzone_Clean$Daily.Max.8.hour.Ozone.Concentration.Clean, start = c(

#Created a monthly time series
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean.Monthly.Ozone, start = c(2010,1), end = c(201
```

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

```r
#11
#Decomposed the daily time series
GaringerOzone.daily.decomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily.decomp)
```




```r
#Decomposed the monthly time series
GaringerOzone.monthly.decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(GaringerOzone.monthly.decomp)
```



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
#Ran a seasonal Mann Kendall test
GaringerOzone.monthly.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

summary(GaringerOzone.monthly.trend)
```

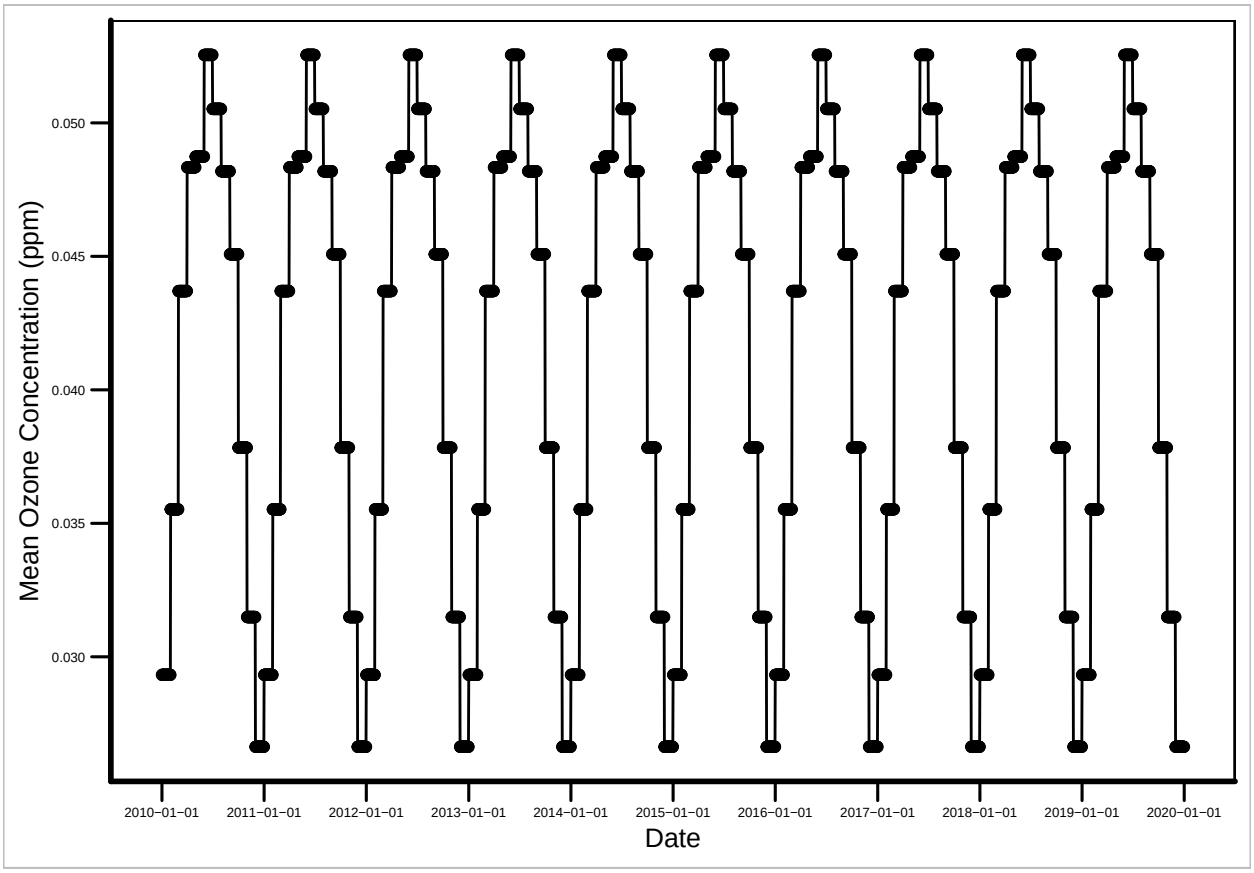
```
## Score =  444 , Var(Score) = 1388
## denominator =  489.653
## tau = 0.907, 2-sided pvalue =< 2.22e-16
```

Answer: The seasonal Mann-Kendall is the most appropriate for this analysis because the data has seasonality, and none of the other tests account for seasonality.

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
#Created a plot with mean monthly ozone
Mean.Monthly.Ozone.Plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean.Monthly.Ozone)) +
  geom_point() +
```

```
  geom_line() +
  scale_x_date(date_breaks="1 year") +
  ylab("Mean Ozone Concentration (ppm)")
print(Mean.Monthly.Ozone.Plot)
```



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: Looking at the graph, it appears that the ozone concentrations change over the seasons yet stay the same over the years, with the same minimums and maximums throughout the seasons each year. These findings align with the results of the seasonal Mann-Kendall test, which had a pvalue =< 2.22e-16, because the p-value is less than 0.05, we can reject the null hypothesis and accept the alternative hypothesis that the ozone concentrations have changed over the 2010s at this station (when we use a test for seasonality).

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
#Created a dataframe with trend and remainder columns, left out seasonal column
```

```
GaringerOzone_Components <- as.data.frame(GaringerOzone.monthly.decomp$time.series[,2:3])

#Made monthly dataframe the same size as the component dataframe
GarginerOzone_Monthly.Comp <- GaringerOzone.monthly %>%
  distinct(Date.MY, .keep_all = TRUE)

#Combined the component dataframe with the monthly dataframe
GaringerOzone_Components <-
  mutate(GaringerOzone_Components,
         Observed = GarginerOzone_Monthly.Comp$Mean.Monthly.Ozone,
         Date = GarginerOzone_Monthly.Comp$Date.MY)

#16
#Ran a time series analysis on the dataframe (without seasonality)
GaringerOzone.monthly.ts <- ts(GaringerOzone_Components$Observed, start = c(2010,1), end = c(2019,12), f

#Ran the Mann Kendall test
GaringerOzone.mk <- Kendall::MannKendall(GaringerOzone.monthly.ts)

summary(GaringerOzone.mk)
```

```
## Score =  -80 , Var(Score) = 192866.7
## denominator =  6864.692
## tau = -0.0117, 2-sided pvalue =0.85724
```

Answer: This Mann Kendall test shows that the ozone concentrations do not change over the years. This Mann Kendall test had a pvalue =0.85724, because the p-value is greater than 0.05, we do not reject the null hypothesis that the ozone concentrations not have changed over the 2010s at this station (when we use a test that doesn't include seasonality). The seasonal Mann Kendall test had a pvalue =< 2.22e-16, and because this p-value was less than 0.05, we rejected the null hypothesis and accepted the alternative hypothesis that the ozone concentrations have changed over the 2010s at this station (when we use a test for seasonality). This makes sense because the ozone concentratuons do not change when we don't consider seasonality, but the ozone concentrations do change when we account for seasonality, which is supported by the results of the Mann Kendall tests.