Assignment 8: Time Series Analysis

Sokna Kry

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 Loading Package
getwd()
```

```
## [1] "/Users/sokna/Documents/EDA-Spring2023"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(zoo)
```

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

```
library(trend)
# Set thene
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.

```
#2 Import the ten datasets from the Ozone_TimeSeries folder


Ozone2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                      stringsAsFactors = TRUE)

Ozone2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                      stringsAsFactors = TRUE)

Ozone2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                      stringsAsFactors = TRUE)

Ozone2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                      stringsAsFactors = TRUE)

Ozone2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                      stringsAsFactors = TRUE)

Ozone2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                      stringsAsFactors = TRUE)
```

```
Ozone2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                      stringsAsFactors = TRUE)

Ozone2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                      stringsAsFactors = TRUE)

Ozone2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                      stringsAsFactors = TRUE)

Ozone2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                      stringsAsFactors = TRUE)

GaringerOzone <- rbind(Ozone2010, Ozone2011, Ozone2012, Ozone2013, Ozone2014, Ozone2015,
                       Ozone2016, Ozone2017, Ozone2018, Ozone2019)
```

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 data column as a date class
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

#4 Wrangle dataset

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

#5 Generate a daily dataset

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

names(Days) <- "Date"


#6 Combine data

GaringerOzone <- left_join(Days, GaringerOzone_Processed, by = "Date")
```

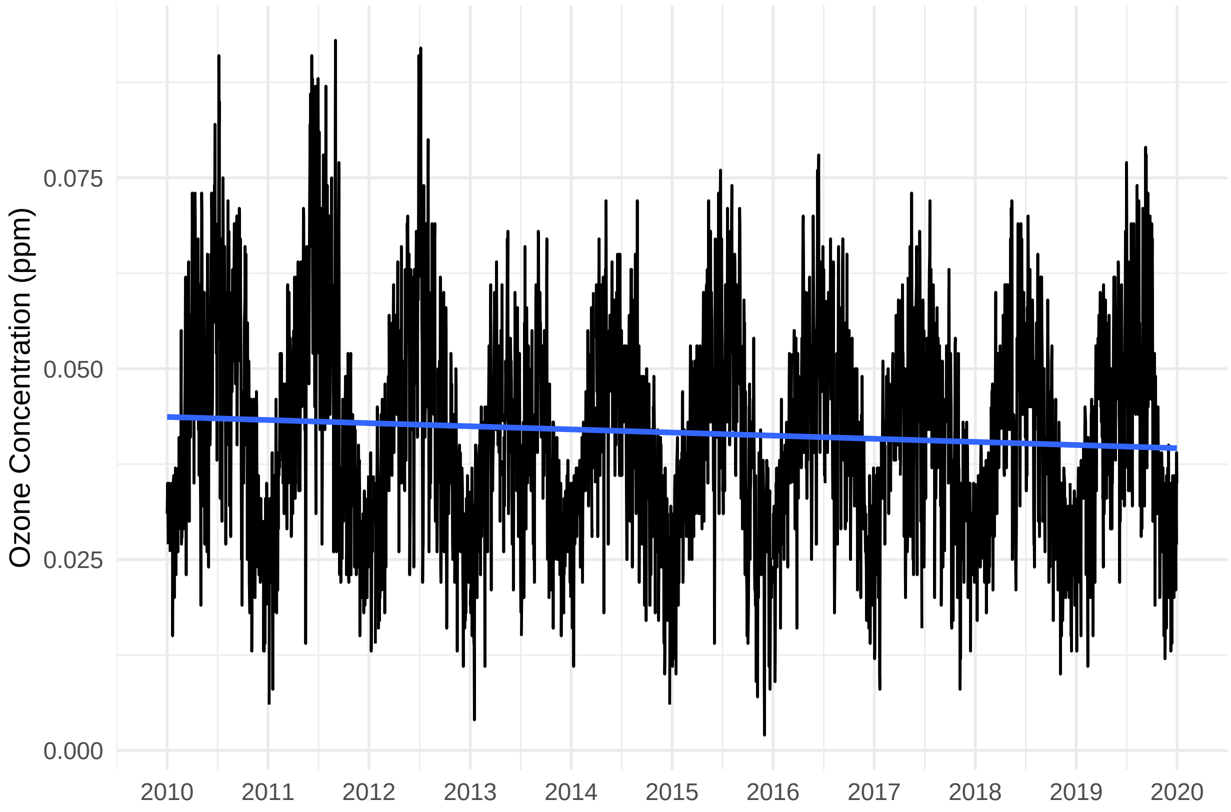
Visualize

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 Create a line plot

ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_y_continuous(name = "Ozone Concentration (ppm)") +
  scale_x_date(name = "", date_breaks = "1 year", date_labels = "%Y") +
  theme_minimal()
```

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

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



Answer: The plot suggest a season 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 Use a linear interpolation to fill in missing daily data for ozone concentration

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

Answer: Linear interpolation is a method of estimating values between two known data points by drawing a straight line between them. It is particularly useful when we have missing data points that are evenly spaced over time. Piecewise constant interpolation, on the other hand, predicts values between the adjacent data points by assuming that the missing values remain constant until the next data point. This method is useful when the missing data points are clustered around a particular time. Spline interpolation is a more complex method that also predicts values between data points by using a mathematical function. It requires more assumptions about the underlying data and takes into account more surrounding data points than linear interpolation. In the case of missing data points in the Ozone concentration data, since the missing values are evenly spaced over time, we can use linear interpolation as a convenient method to estimate the values between the known data points.

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 Create a new data frame called `GaringerOzone.monthly`

GaringerOzone.monthly <- GaringerOzone_clean %>%
  mutate(year = year(Date), month = month(Date)) %>%
  group_by(year, month) %>%
  summarise(mean_ozone_concentration = mean(Ozone.Concentration, na.rm = TRUE))
```

```
## ‘summarise()‘ has grouped output by ’year’. You can override using the
## ‘.groups‘ argument.
```

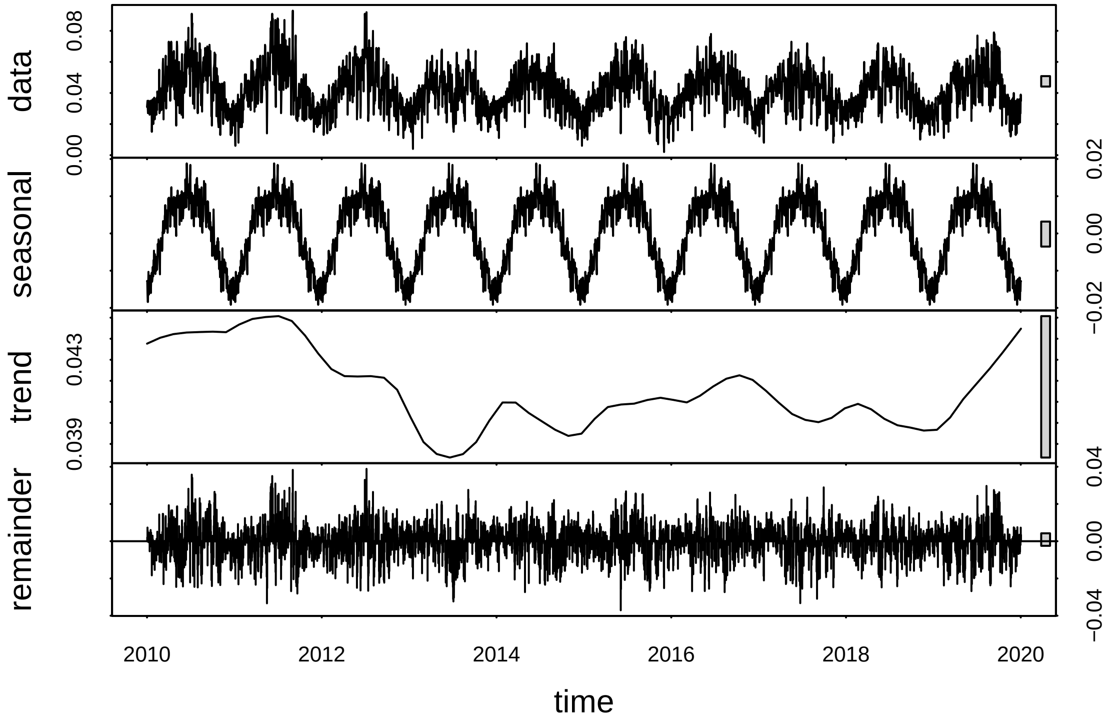
```
# Create a new Date column with each month-year combination being set as the first day of the month
GaringerOzone.monthly$Date <- as.Date(paste0(GaringerOzone.monthly$year, "-",
                                             GaringerOzone.monthly$month, "-01"))
```

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.

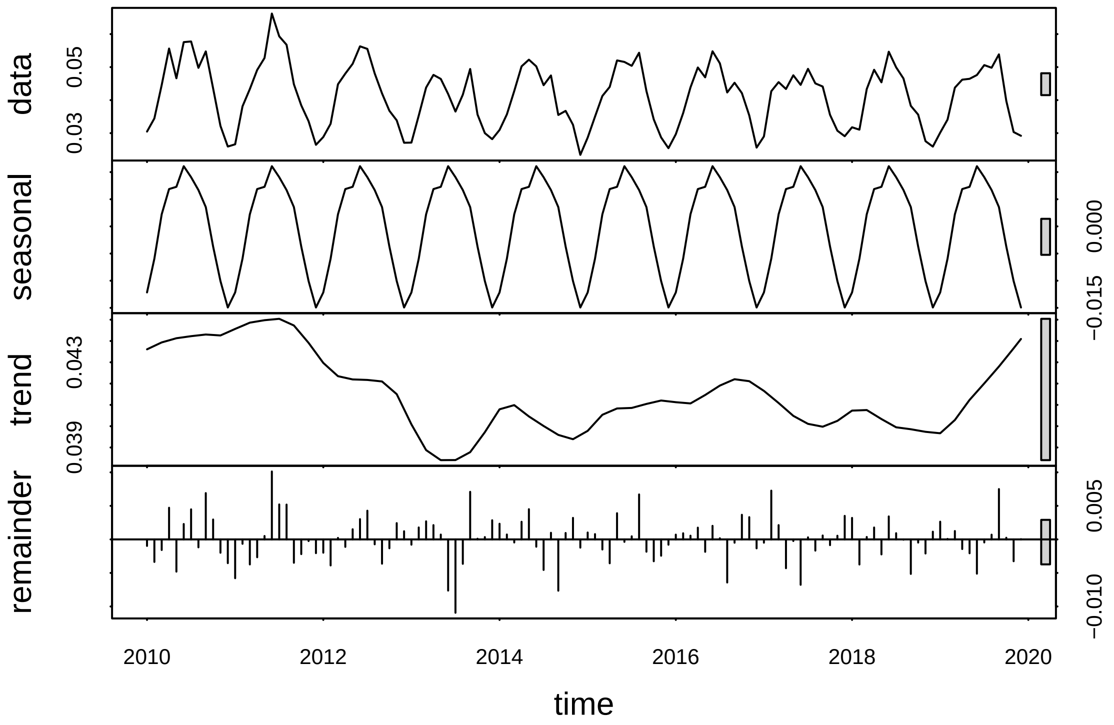
```
#10 Generate two time series objects

GaringerOzone.daily.ts<-ts(GaringerOzone_clean$Ozone.Concentration, start = c(2010,1),
                           frequency = 365)

GaringerOzone.monthly.ts<-ts(GaringerOzone.monthly$mean_ozone_concentration, 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 Decompose the daily and the monthly time series objects and plot them

GaringerOzone.Daily.Decomp<- stl(GaringerOzone.daily.ts,s.window="periodic")

plot(GaringerOzone.Daily.Decomp)
```



```
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 Run a monotonic trend analysis for the monthly Ozone series

library(Kendall)

# Run Mann-Kendall test on monthly Ozone series
Ozone_trend <- MannKendall(GaringerOzone.monthly.ts)

# Inspect results
summary(Ozone_trend)
```

```
## Score =  -424 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.0594, 2-sided pvalue =0.33732
```

```
# Run SMK test
MonthlyOzone_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Inspect results

summary(MonthlyOzone_trend1)
```

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

```
MonthlyOzone_trend2 <- trend::smk.test(GaringerOzone.monthly.ts)

# Inspect results

summary(MonthlyOzone_trend2)
```

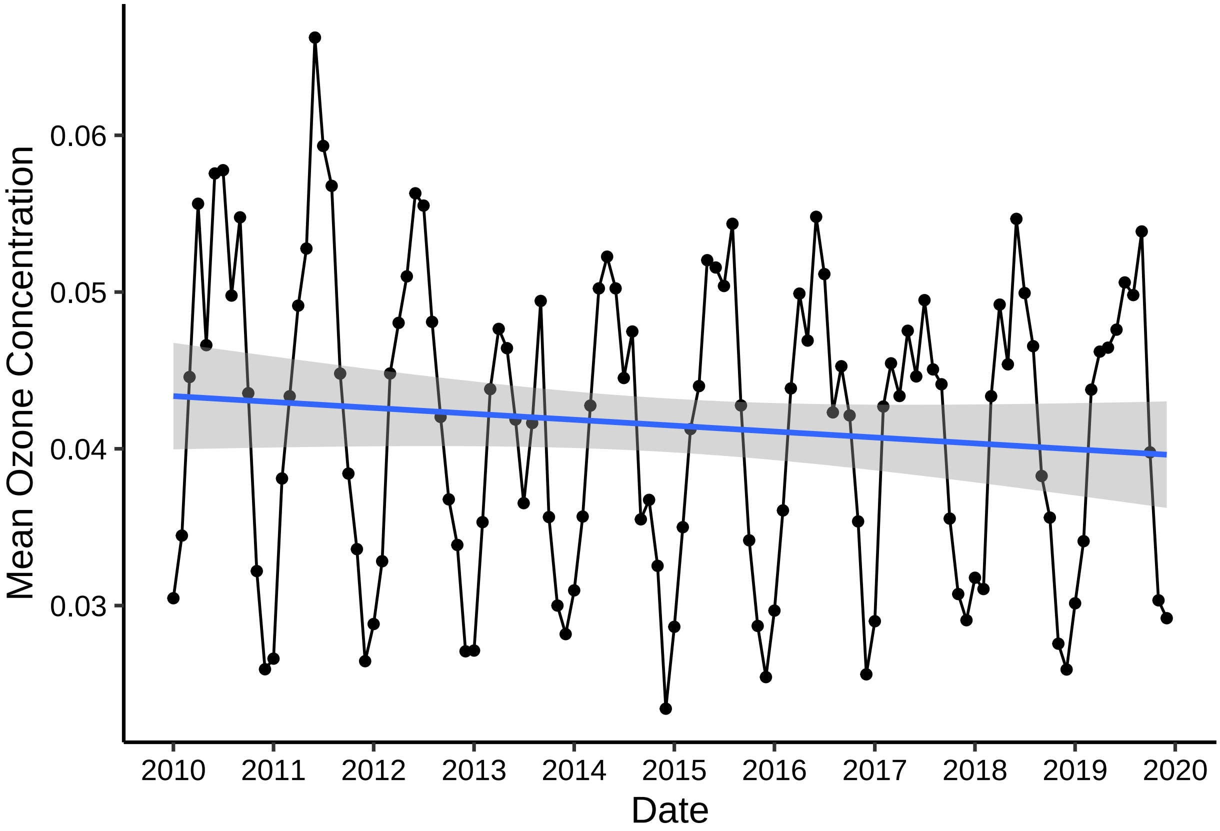
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                      S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 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: The Seasonal Mann-Kendall test is the most appropriate for this analysis as it considers the seasonal pattern in the data. This test is specifically designed to detect trends that follow a consistent pattern over time, while also taking into account the seasonal fluctuations of the data.

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 Plot depicting mean monthly ozone concentrations over time

MonthlyOzone_plot <- ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_ozone_concentration)) +
  geom_point() +
  geom_line() +
  ylab("Mean Ozone Concentration") +
  xlab("Date") + # Add x-axis label
  scale_x_date(date_breaks = "1 year",date_labels = "%Y") +
  geom_smooth(method = lm)

print(MonthlyOzone_plot)
```

```
## `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: According to the graph, we can see the slighly trend of ozone (decrease over time). By looking at the statistical test, the result is aligned with what we see from the graph. z-value of the test is negative, which indicates a decreasing trend in the data over time. The p-value of the test is 0.046, which is less than 0.05. This indicates that we can reject the null hypothesis that there is no trend in the data and conclude that there is evidence of a statistically significant trend 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

# We can extract the components and turn them into data frames
GaringerOzone_Components <- as.data.frame(GaringerOzone.Monthly.Decomp$time.series[,1:3])

GaringerOzone_Components <- mutate(GaringerOzone_Components,
        Observed = GaringerOzone.monthly$mean_ozone_concentration,
        Date = GaringerOzone.monthly$Date)

#16 Run the Mann Kendall test on the non-seasonal Ozone monthly series


Ozone.monthly_components.ts<-ts(GaringerOzone_Components$Observed, start=c(2010,1),
                             frequency=12)

Ozone.monthly_components_trend1 <- Kendall::MannKendall(Ozone.monthly_components.ts)

Ozone.monthly_components_trend1
```

```
## tau = -0.0594, 2-sided pvalue =0.33732
```

```
summary(Ozone.monthly_components_trend1)
```

```
## Score =  -424 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.0594, 2-sided pvalue =0.33732
```

```
Ozone.monthly_components_trend2 <-trend::smk.test(Ozone.monthly_components.ts)

Ozone.monthly_components_trend2
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  Ozone.monthly_components.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

```
summary(Ozone.monthly_components_trend2)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: Ozone.monthly_components.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: The non seasonal plot and season plot does not show much different trend; they both show the slight change of trend over time.