Assignment 8: Time Series Analysis

David Robinson

Fall 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

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

```
## here() starts at C:/Users/dhr20/OneDrive - Duke University/1 - Academics/1 - First Year/1 - Fall 202
```

```
here()
```

```
## [1] "C:/Users/dhr20/OneDrive - Duke University/1 - Academics/1 - First Year/1 - Fall 2023/2 - Enviro
```

```
#install.packages("tidyverse")
library(tidyverse)
```

```
## -- 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 1.9.2     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
```

```
#install.packages("lubridate")
library(lubridate)
#install.packages("zoo")
library(zoo)
```

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

```
#install.packages("trend")
library(trend)
#install.packages("dplyr")
library(dplyr)

#Set 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

#Import datasets
Air_2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                         stringsAsFactors = TRUE)
Air_2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                         stringsAsFactors = TRUE)
Air_2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                         stringsAsFactors = TRUE)
Air_2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                         stringsAsFactors = TRUE)
Air_2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                         stringsAsFactors = TRUE)
Air_2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                         stringsAsFactors = TRUE)
Air_2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                         stringsAsFactors = TRUE)
Air_2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                         stringsAsFactors = TRUE)
```

```
Air_2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                         stringsAsFactors = TRUE)
Air_2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                         stringsAsFactors = TRUE)

GaringerOzone <- rbind(Air_2010, Air_2011, Air_2012, Air_2013, Air_2014,
                       Air_2015, Air_2016, Air_2017, Air_2018, Air_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.

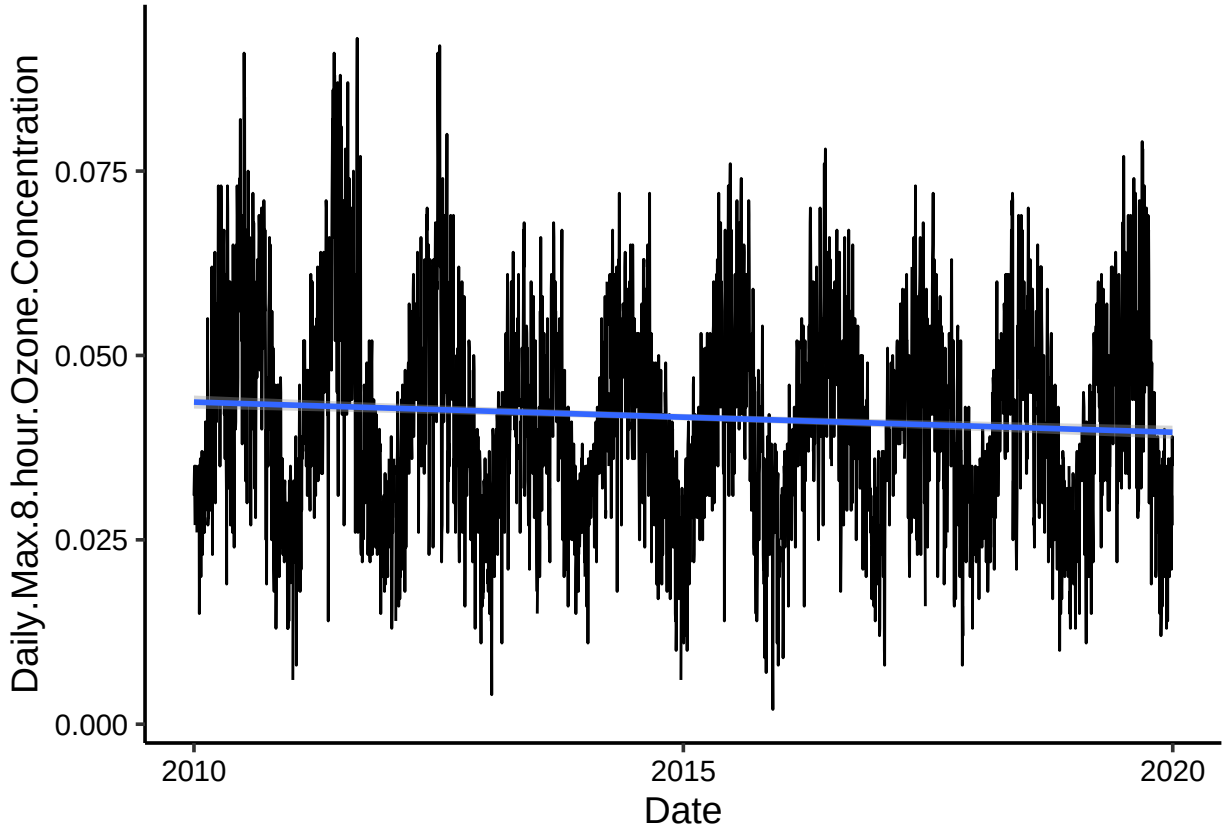
```
#3

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

#4

#Wrangle dataset
GaringerOzone_wrangled <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration,
         DAILY_AQI_VALUE)

#5

#Create new data frame
Days <- as.data.frame(seq(ymd("2010-01-01"),ymd("2019-12-31"), by = "days"))
colnames(Days)[1] = "Date"

#6

#Combine the data frames
GaringerOzone <- left_join(Days,GaringerOzone_wrangled, 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

#visualize trend via line plot
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = lm)
```

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

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



Answer: The plot suggests a trend that ozone concentration has decreased steadily 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

#Check number of NA's
summary(GaringerOzone) #NA's = 63
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
##                       NA's   :63                           NA's   :63
```

```
#Use linear interpolation to fill in missing daily data
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <- zoo::na.approx(GaringerOzone$Daily.Max.8.hour.Ozo

#Check number of NA's again
summary(GaringerOzone) #NA's = 0
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04151                      Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
##                                                            NA's   :63
```

Answer: The periods of missing data are short – therefore we can use linear interpolation.

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 new data frame for monthly data
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(Month = month(ymd(GaringerOzone$Date))) %>%
  mutate(Year = year(ymd(GaringerOzone$Date))) %>%
  mutate(Date = my(paste0(Month,"-",Year))) %>%
  group_by(Date) %>% #QUESTION -- clarify here
  summarize(Monthly.Max.8.hour.Ozone.Concentration = mean(Daily.Max.8.hour.Ozone.Concentration))
```

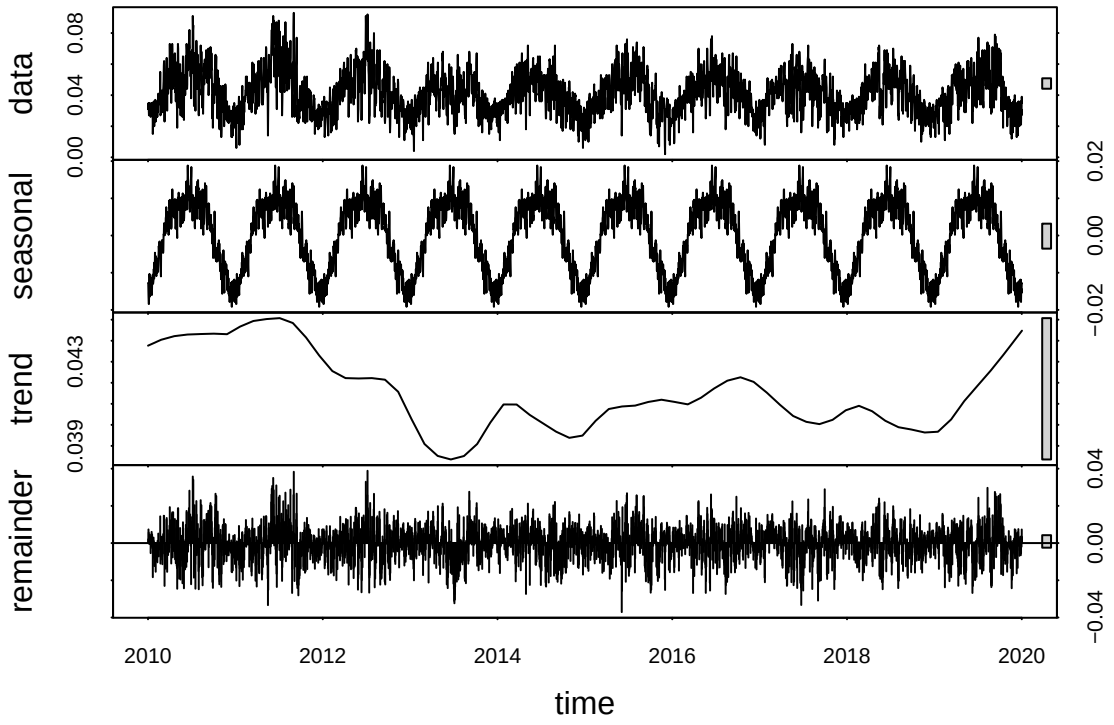
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of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

```
#10

#Define first day, month, year
f_day = day(first(GaringerOzone$Date))
f_month = month(first(GaringerOzone.monthly$Date))
f_year = year(first(GaringerOzone$Date))
```

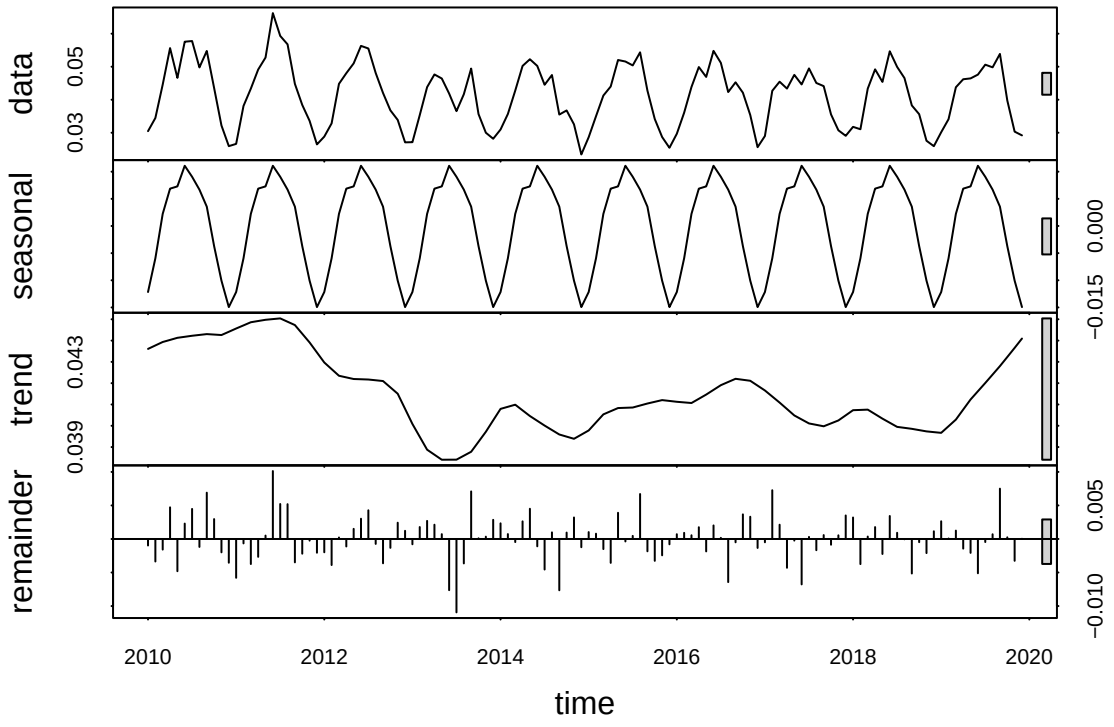
```r
#Create daily time series object
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(f_year,f_day),
                             frequency = 365)

#Create monthly time series object
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Monthly.Max.8.hour.Ozone.Concentration,
                               start = c(f_year,f_month),
                               frequency = 12)
```

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

```r
#11

#Decompose the daily time series object
GaringerOzone.daily.ts.decomposed <-
  stl(GaringerOzone.daily.ts,
      s.window = "periodic")

#Plot the component
plot(GaringerOzone.daily.ts.decomposed)
```



```r
#Decompose the monthly time series object
GaringerOzone.monthly.ts.decomposed <-
  stl(GaringerOzone.monthly.ts,
      s.window = "periodic")

#Plot the component
plot(GaringerOzone.monthly.ts.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?

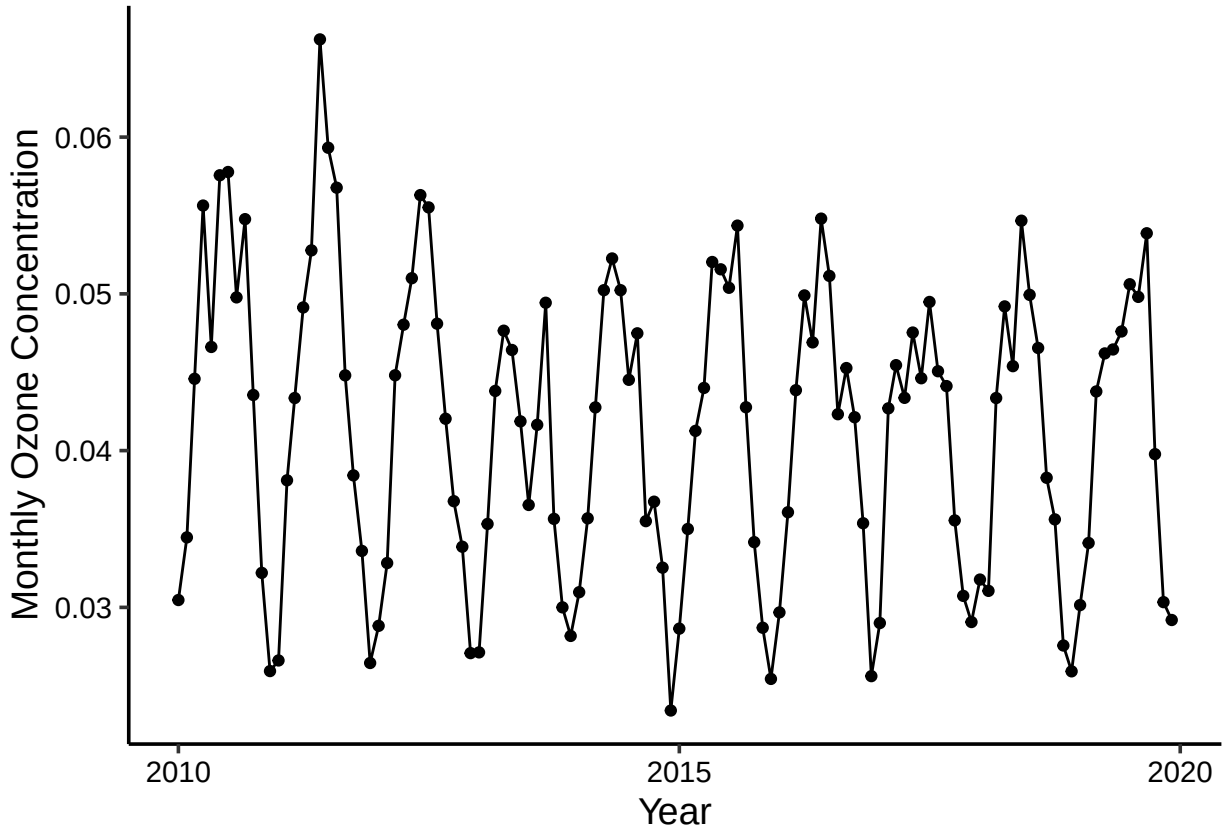
```r
#12

#Run monotonic trend analysis
GaringerOzone.monthly.ts.trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

#Show analysis
summary(GaringerOzone.monthly.ts.trend1)
```

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

Answer: The Mann-Kendall test is used when seasonanility is not expected to be present or when trends occur in different directions in different seasons. The H0 for the Mann-Kendall test is that the time series is stationary. The HA is that the time series does follow a seasonal trend.

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

#Create plot depicting monthly concentrations over time
GaringerOzone.monthly.ts.plot <-
  ggplot(GaringerOzone.monthly, aes(x = Date,
                                    y = Monthly.Max.8.hour.Ozone.Concentration)) +
  geom_point() +
  geom_line() +
  labs(y = "Monthly Ozone Concentration", x = "Year")

#Show plot
print(GaringerOzone.monthly.ts.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: Given a p-value of less than 0.05 at 0.046724, we reject H0 and thus conclude that there is a monotonic, seasonal trend to the data. Because tau is negative, we conclude that the trend is decreasing. Score = -77 , Var(Score) = 1499 denominator = 539.4972 tau = -0.143, 2-sided pvalue =0.046724

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

#Slice data frame
GaringerOzone.monthly.ts_Components <-
  as.data.frame(GaringerOzone.monthly.ts.decomposed$time.series[,1:3])

#Subtract components
GaringerOzone.monthly.ts_Components <-
  mutate(GaringerOzone.monthly.ts_Components,
         Observed =
           GaringerOzone.monthly$Monthly.Max.8.hour.Ozone.Concentration,
         Date = GaringerOzone.monthly$Date,
         Nonseason = Observed - seasonal)

#Cast as time series
GaringerOzone.monthly.ts_Components_ts <- ts(GaringerOzone.monthly.ts_Components$Nonseason,
                                start = c(f_year,f_month),
                                frequency = 12)

#16

#Run monotonic trend analysis
GaringerOzone.monthly.ts.trend2 <- Kendall::MannKendall(GaringerOzone.monthly.ts_Components_ts)

#Show trend analysis
summary(GaringerOzone.monthly.ts.trend2)
```

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

Answer: Given a p-value of less than 0.05 at 0.0075402, we reject H0 and thus conclude that there is a monotonic, seasonal trend to the data. Because this p-value is less than before, we conclude that there is a stronger trend when we extract the seasonal component from the data. Because tau is negative, we conclude that the trend is decreasing. Score = -1179 , Var(Score) = 194365.7 denominator = 7139.5 tau = -0.165, 2-sided pvalue =0.0075402