Assignment 7: Time Series Analysis

Blair Johnson

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on time series analysis.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A07_TimeSeries.Rmd") prior to submission.

The completed exercise is due on Monday, March 14 at 7:00 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

```
#1
getwd()
```

```
## [1] "Z:/ENV872/Environmental_Data_Analytics_2022"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- 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)
library(ggplot2)

mytheme <- theme_bw(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")
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

ozone.2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
ozone.2011 <-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
ozone.2012 <-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
ozone.2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
ozone.2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
ozone.2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
ozone.2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
ozone.2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
ozone.2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
ozone.2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")

GaringerOzone <-rbind(ozone.2010, ozone.2011, ozone.2012, ozone.2013, ozone.2014, ozone.2015, ozone.201
```

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$Date <- mdy(GaringerOzone$Date)

# 4
```

```
GaringerOzone.filtered <- GaringerOzone %>%
  select(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 = "days"))
names(Days) <- c("Date")

# 6

GaringerOzone <- left_join(Days, GaringerOzone.filtered)
```

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

```
names(GaringerOzone) <- c("Date","Ozone", "AQI")
```

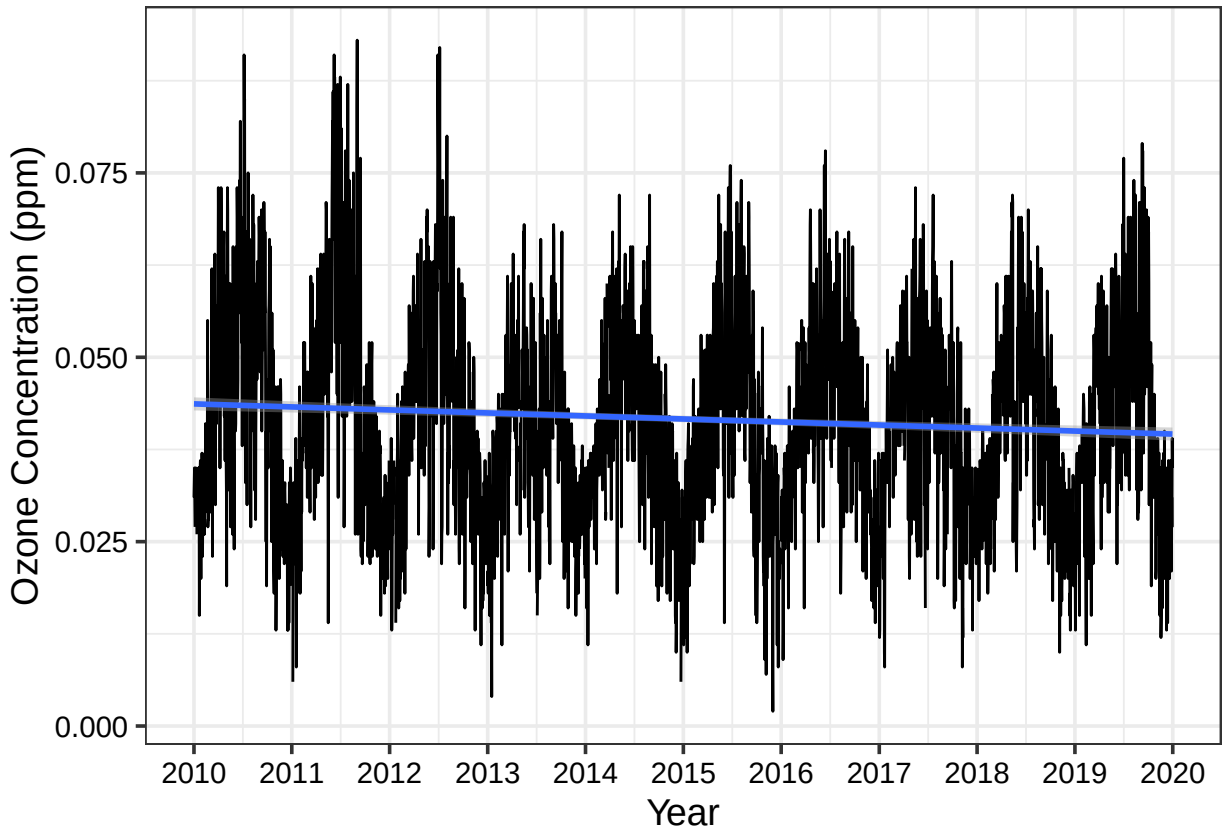
Visualize

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
GaringerOzone.plot <- ggplot(GaringerOzone, aes(y=Ozone, x=Date)) +
  geom_line() +
  geom_smooth(method = "lm") +
 scale_x_date(limits = as.Date(c("2010-01-01", "2019-12-31")),
    date_breaks = "1 year", date_labels = "%Y") +
  labs(y ="Ozone Concentration (ppm)", x= "Year")
print(GaringerOzone.plot)
```

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

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



Answer: Based on this time series plot, the ozone concentrations are higher in 2010-2012 than in the following years up until 2010, then decreases following that year. This plot also shows that the ozone concentrations increase and decrease based on the seasons because higher ozone concentrations are more prevalent in the middle of each year and are at their lowest at the end of each year. Based on the trendline, there is a slight negative trend 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
summary(GaringerOzone)
```

```
##       Date                Ozone              AQI
##  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
```

```
Garinger.data.clean <-
  GaringerOzone %>%
  mutate(Ozone.clean = zoo::na.approx(Ozone))
```

Answer: The linear interpolation allows us to fill in missing data using the daily data that falls before and after the missing period. Since the spline interpolation relies on a quadratic formula, using that interpolation method could yield daily data that does not accurately represent the dataset. We do not use the piecewise constant because it is not ideal for a continuous dataset(like the one in this exercise) because it draws from the nearest neighbor and does not fill in gaps as neatly.

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 <- Garinger.data.clean %>%
  mutate(Month =month(Date),
         Year=year(Date)) %>%
         mutate(Month_Year=my(paste0(Month, "-", Year))) %>%
         group_by(Month_Year) %>%
  summarise(MeanOzone = mean(Ozone.clean))


head(GaringerOzone.monthly, 6)
```

```
## # A tibble: 6 x 2
##   Month_Year MeanOzone
##   <date>          <dbl>
## 1 2010-01-01    0.0305
## 2 2010-02-01    0.0345
## 3 2010-03-01    0.0446
## 4 2010-04-01    0.0556
## 5 2010-05-01    0.0466
## 6 2010-06-01    0.0576
```

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.

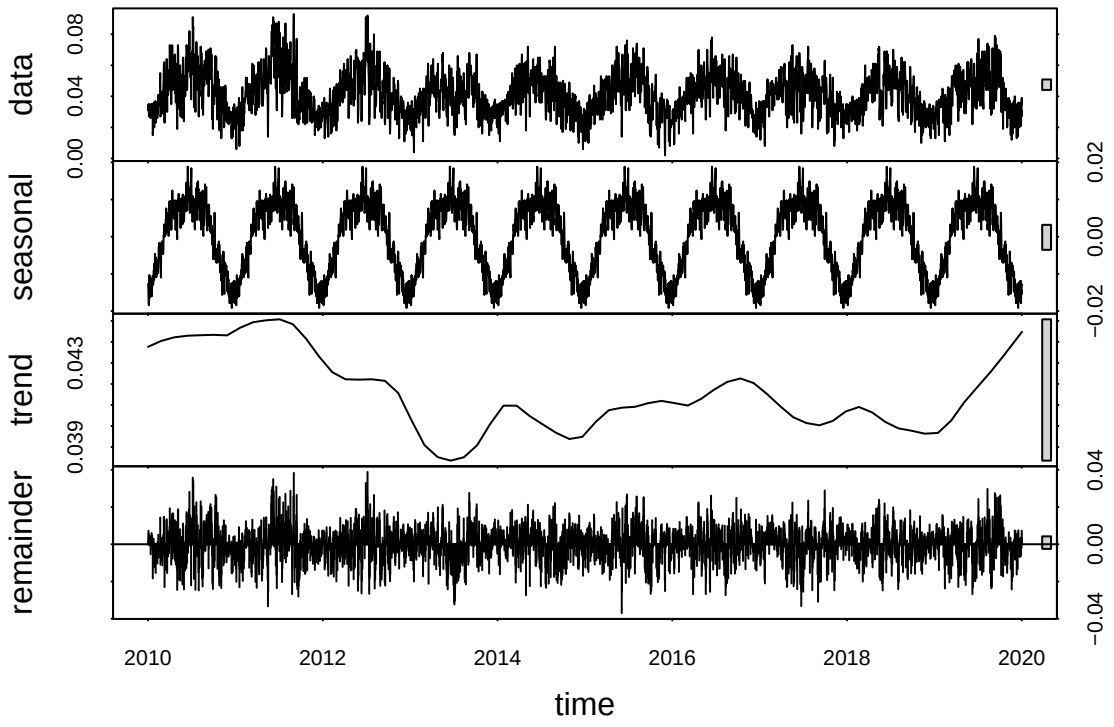
```
#10
GaringerOzone.daily.ts <- ts(Garinger.data.clean$Ozone.clean, start= c(2010, 1), frequency = 365)

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

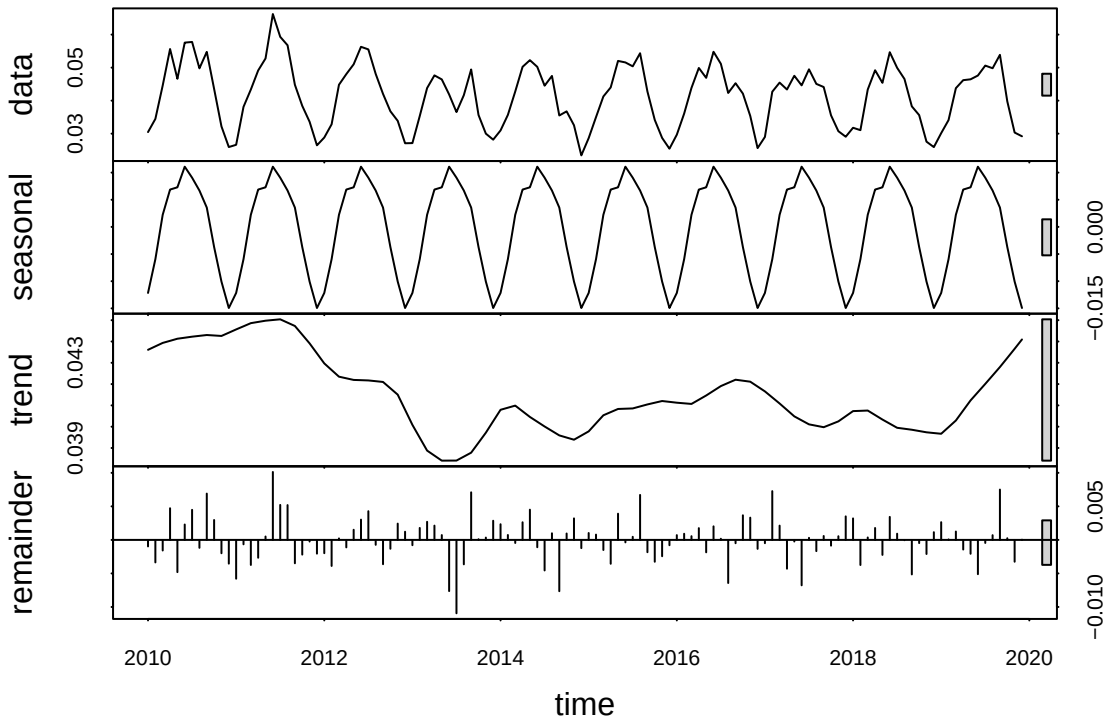
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11

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




```
Garinger.monthly.decomposed <-stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(Garinger.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.trend.2<-trend::smk.test(GaringerOzone.monthly.ts)
summary(GaringerOzone.monthly.trend.2)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
```

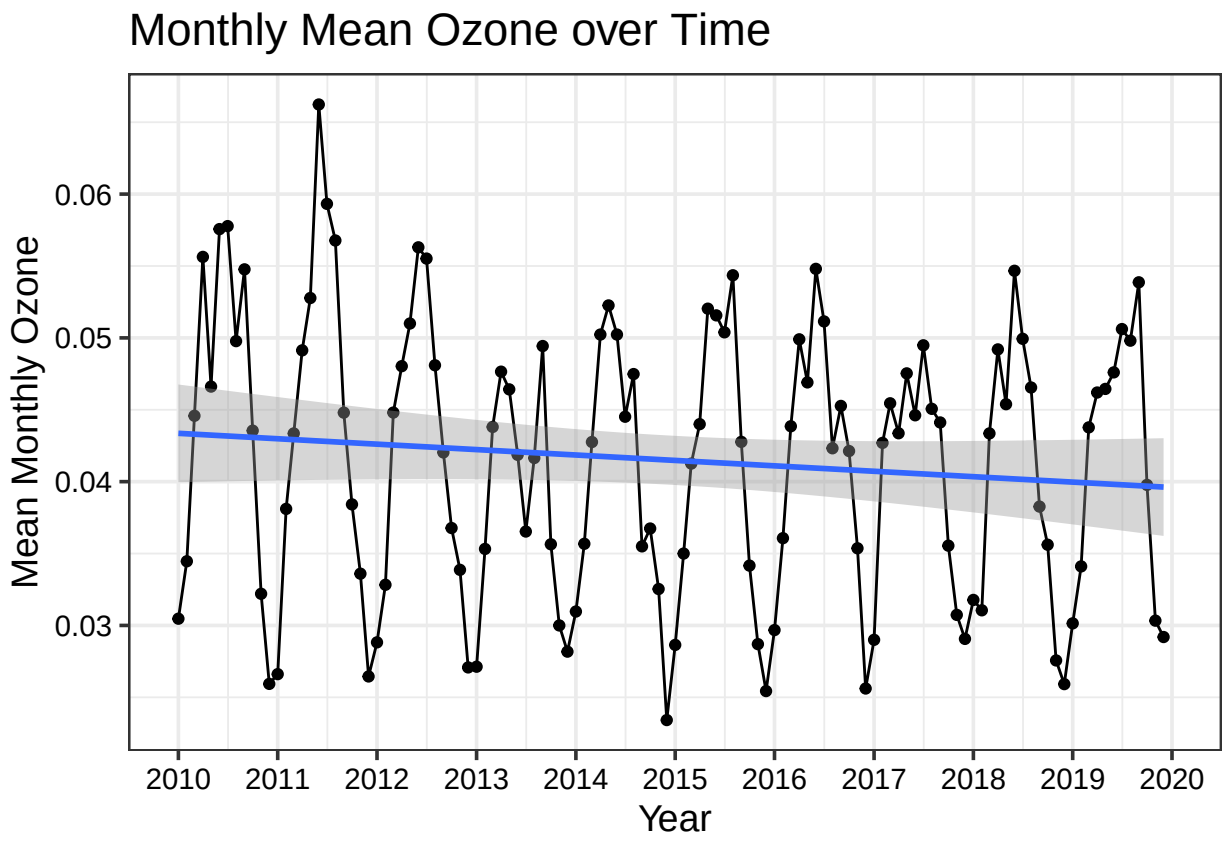
```
## H0
##                      S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 2:   S = 0   -1  125 -0.022  0.000  1.00000
## Season 3:   S = 0   -4  124 -0.090 -0.269  0.78762
## Season 4:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 5:   S = 0  -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 is most appropriate because it is appropriate for data that shows seasonal trends. Based off of our analysis, we see that the ozone levels in the dataset change seasonally; therefore, the seasonal Mann-Kendall accounts for those changes.

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.Plot <- ggplot(GaringerOzone.monthly, aes(x=Month_Year, y=MeanOzone)) + geom_poin
    date_breaks = "1 year", date_labels = "%Y")
print(GaringerOzone.Monthly.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: Based on the results in this graph, the year 2011 has the highest monthly mean ozone and the monthly mean ozone is at its lowest in late 2014. The decomposed plot also highlights the trends in the ozone concentration by illustrating a sharp decrease in ozone concentrations in late 2014 followed by an increase in 2015 and slight increases and decreases up until 2019 where the ozone concentration begins to sharply increase (Statistical test output: p-value =0.046724, Score = -77, Var(Score) =1499). Therefore, the p-value shows these results are statistically significant and the results satisfy the research question that the ozone concentrations change over time starting in the 2010.

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

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

GaringerOzone.nonseasonal.monthly.ts <- mutate(GaringerOzone.nonseasonal.monthly.ts,
        Observed = GaringerOzone.monthly$MeanOzone,
        Date = GaringerOzone.monthly$Month_Year)

GaringerOzone.nonseasonal.monthly.ts <-
```

```
      mutate (GaringerOzone.nonseasonal.monthly.ts,
              Observed.Minus.Seasonal = GaringerOzone.nonseasonal.monthly.ts$Observed - GaringerOzone.
#16

GaringerOzone.nonseasonal.ts <- ts(GaringerOzone.nonseasonal.monthly.ts$Observed.Minus.Seasonal, start =


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

GaringerOzone.monthly.nonseasonal.trend
```

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

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

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

Answer: When comparing the nonseasonal data, we obtain a smaller p-value of 0.0075402 and a score of -1179. Therefore, this p-value is statistically significant and shows that seasonality plays a large role in the level of ozone concentrations over time.