Assignment 8: Time Series Analysis

Danielle Butler

Spring 2025

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 and 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 errors
```

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

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

```
library(here)
```

```
## here() starts at /home/guest/EDA_Spring2025
```

```
here()
```

```
## [1] "/home/guest/EDA_Spring2025"
```

```
#Set ggplot 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

EPAair_O3_2010 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"), stringsAs
EPAair_O3_2011 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"), stringsAs
EPAair_O3_2012 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"), stringsAs
EPAair_O3_2013 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"), stringsAs
EPAair_O3_2014 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"), stringsAs
EPAair_O3_2015 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"), stringsAs
EPAair_O3_2016 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"), stringsAs
EPAair_O3_2017 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"), stringsAs
EPAair_O3_2018 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"), stringsAs
EPAair_O3_2019 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"), stringsAs

GaringerOzone <- EPAair_O3_2010 %>%
  full_join(EPAair_O3_2011) %>%
  full_join(EPAair_O3_2012) %>%
  full_join(EPAair_O3_2013) %>%
  full_join(EPAair_O3_2014) %>%
  full_join(EPAair_O3_2015) %>%
  full_join(EPAair_O3_2016) %>%
  full_join(EPAair_O3_2017) %>%
  full_join(EPAair_O3_2018) %>%
  full_join(EPAair_O3_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
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

#4
GaringerOzone_processed <- 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 = "day"))
colnames(Days) <- "Date"

#6
GaringerOzone <- left_join(Days,GaringerOzone_processed)
```

```
## Joining with `by = join_by(Date)`
```

Visualize

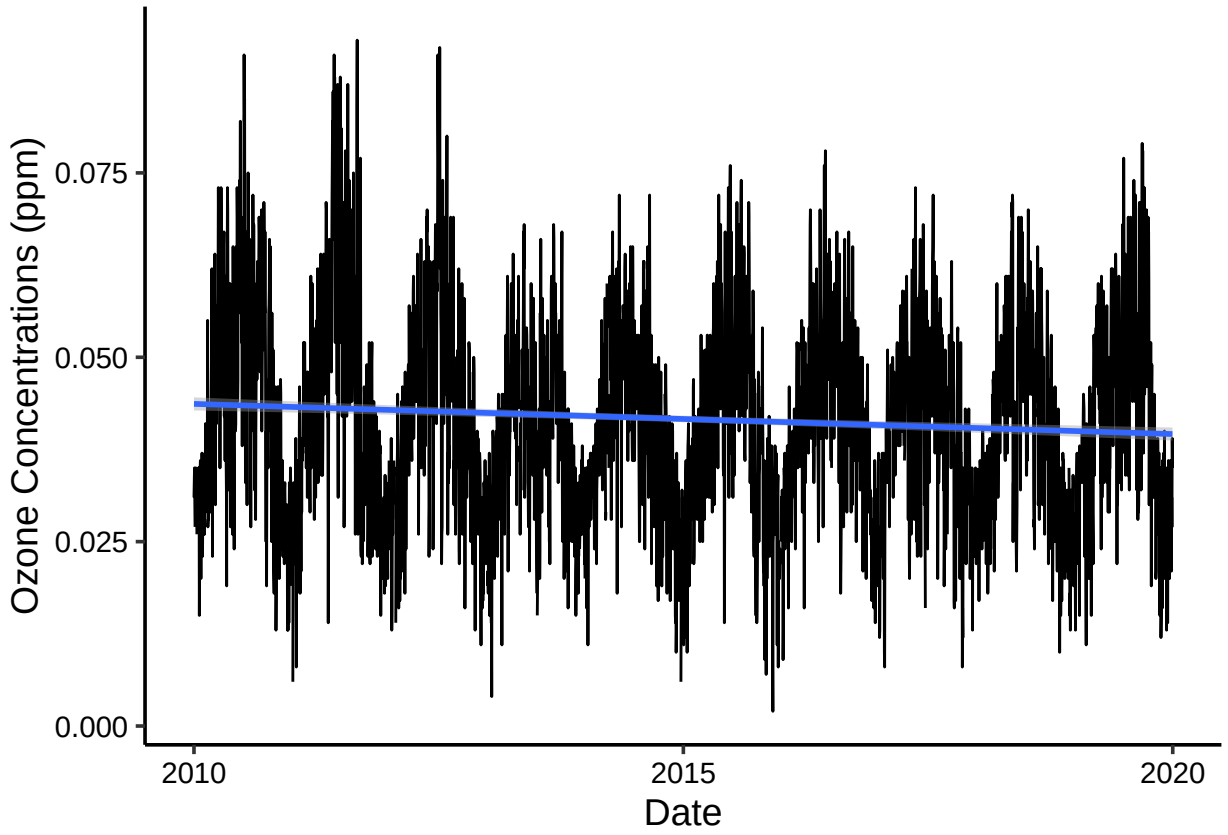
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7
data_plot <-
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth( method = lm )+
  ylab("Ozone Concentrations (ppm)")

print(data_plot)
```

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

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



Answer: The plot shows a decrease in Ozone concentrations 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
Ozone_data <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentrati

summary(Ozone_data)
```

```
##       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
```

```
##  Daily.Max.8.hour.Ozone.Concentration.clean
##  Min.   :0.00200
##  1st Qu.:0.03200
##  Median :0.04100
##  Mean   :0.04151
##  3rd Qu.:0.05100
##  Max.   :0.09300
##
```

Answer: Based on the linear trend we found, linear made the most sense to "connect the dots" of the data missing.

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 <- Ozone_data %>%
  mutate(year = year(Date), month = month(Date)) %>%
  group_by(year, month) %>%
  summarize(mean_ozone = mean(Daily.Max.8.hour.Ozone.Concentration.clean, na.rm = TRUE)) %>%
  ungroup()
```

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

```
#View the result
head(GaringerOzone.monthly)
```

```
## # A tibble: 6 x 3
##    year month mean_ozone
##   <dbl> <dbl>      <dbl>
## 1  2010     1     0.0305
## 2  2010     2     0.0345
## 3  2010     3     0.0446
## 4  2010     4     0.0556
## 5  2010     5     0.0466
## 6  2010     6     0.0576
```

```
#Create a new Date column with the first day of each month
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = make_date(year, month, 1))

#View the result
head(GaringerOzone.monthly)
```

```
## # A tibble: 6 x 4
##    year month mean_ozone Date
##   <dbl> <dbl>      <dbl> <date>
## 1  2010     1     0.0305 2010-01-01
```

```
## 2  2010     2     0.0345 2010-02-01
## 3  2010     3     0.0446 2010-03-01
## 4  2010     4     0.0556 2010-04-01
## 5  2010     5     0.0466 2010-05-01
## 6  2010     6     0.0576 2010-06-01
```

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
#daily time series
GaringerOzone.daily.ts <- ts(Ozone_data$Daily.Max.8.hour.Ozone.Concentration.clean,
    start = c(year(min(Ozone_data$Date)), month(min(Ozone_data$Date))),
    end = c(year(max(Ozone_data$Date)), month(max(Ozone_data$Date))),
                              frequency = 365)

#View the daily time series
head(GaringerOzone.daily.ts)
```

```
## [1] 0.031 0.033 0.035 0.031 0.027 0.030
```

```
#monthly time series
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone,
    start = c(year(min(GaringerOzone.monthly$Date)), month(min(GaringerOzone.monthly$Date))),
    end = c(year(max(GaringerOzone.monthly$Date)), month(max(GaringerOzone.monthly$Date))),
                              frequency = 12)

#View the monthly time series
head(GaringerOzone.monthly.ts)
```

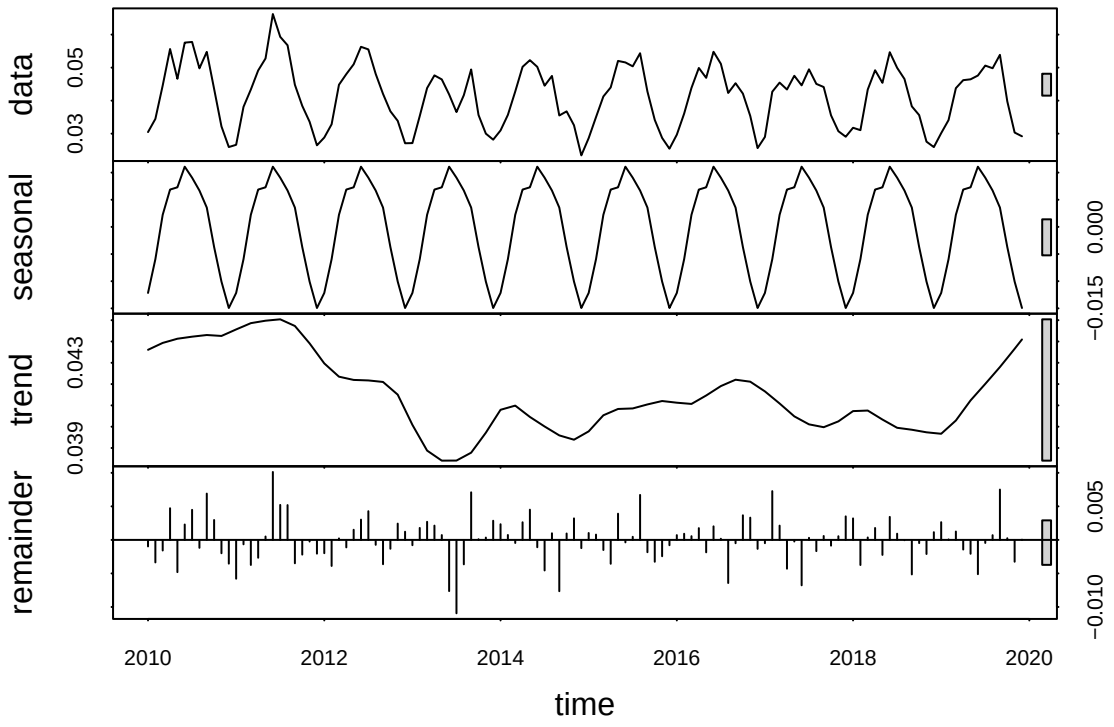
```
## [1] 0.03046774 0.03446429 0.04458065 0.05563333 0.04661290 0.05756667
```

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

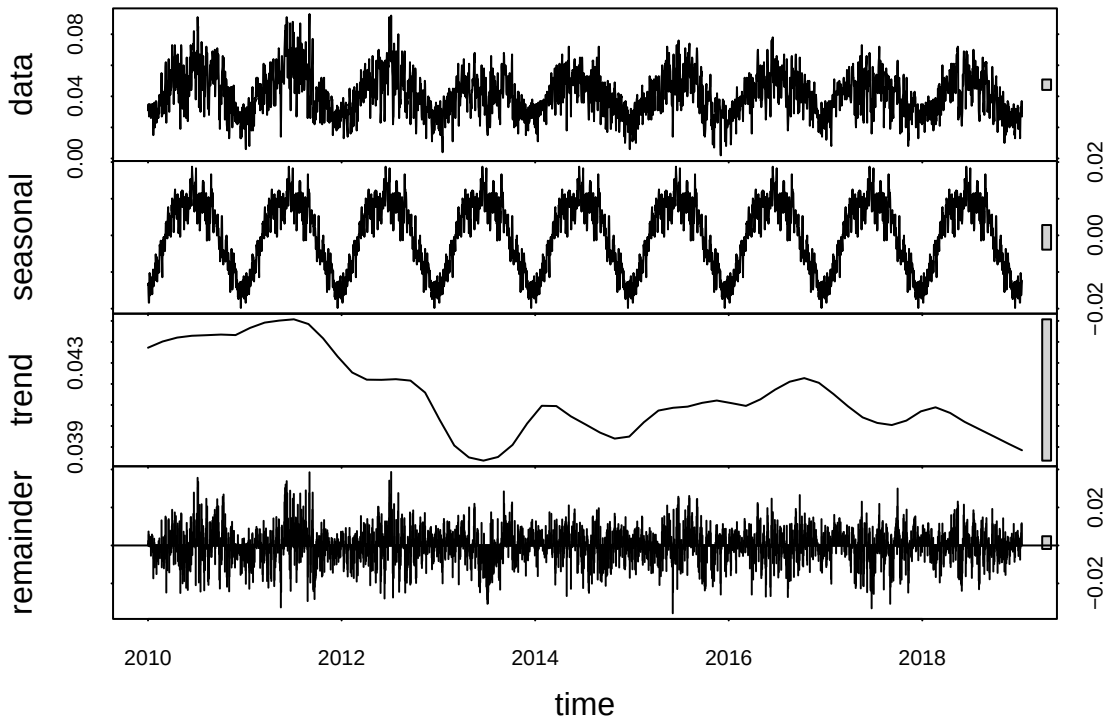
```
#11

# Generate the decomposition
GaringerOzone.monthly.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
GaringerOzone.daily.decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")

#Visualize
plot(GaringerOzone.monthly.decomposed)
```




```
plot(GaringerOzone.daily.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

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

# Inspect results
Ozone_data_trend
```

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

```
summary(Ozone_data_trend)
```

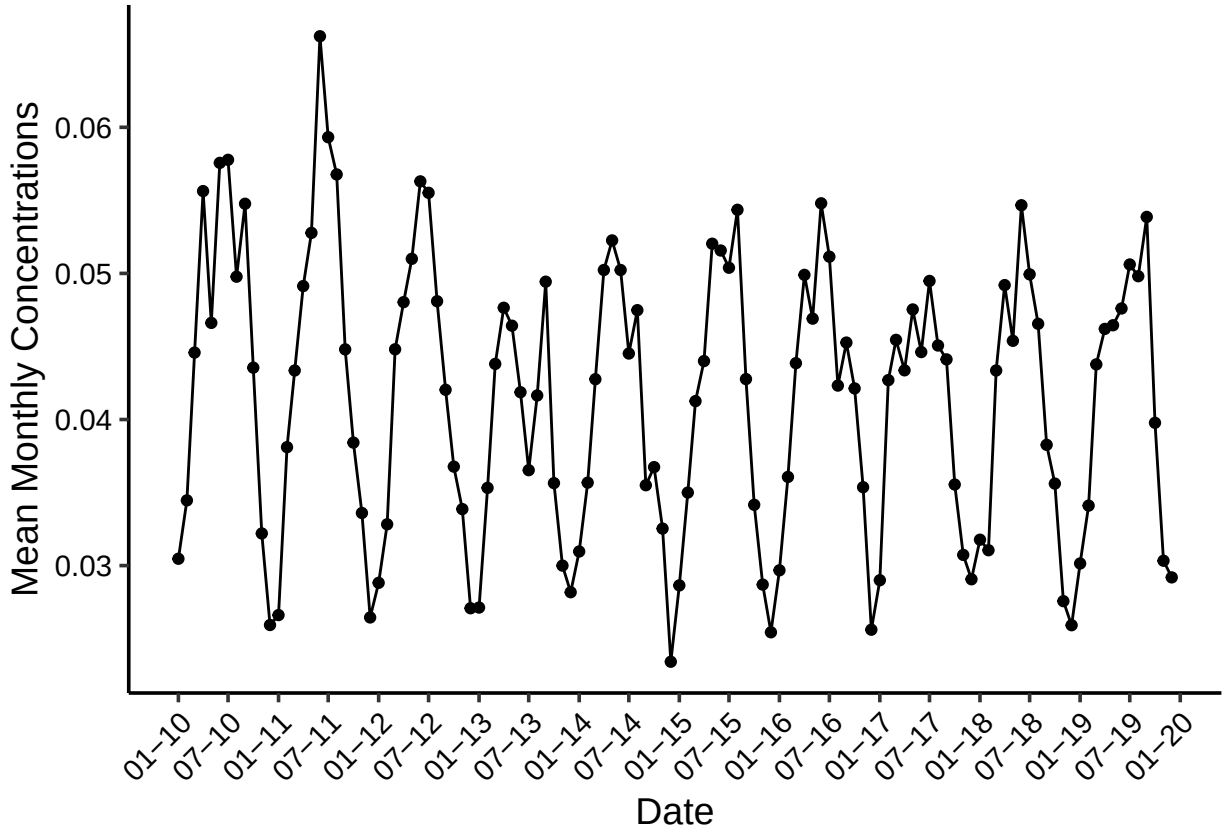
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: There is seasonality to the monthy data so this test will address the seasonality by performing the tests on each season separately and combining the results, helpful for Ozone data, which can change based on season/climate and weather.

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.

```r
# 13
#Visualization
Ozone_data_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_ozone)) +
  geom_point() +
  geom_line() +
  ylab("Mean Monthly Concentrations") +
  scale_x_date(date_breaks = "6 month", date_labels = "%m-%y") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))

# To view the plot
print(Ozone_data_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: Study Question was "have ozone concentrations changed over the 2010s at this station?" In short, mean monthly concentrations change based on the season of this area. In the summer months, there are peaks of mean monthly Ozone data, and in the winter there are valleys. In general, there is a trend of decreasing mean monthly ozone concentrations. The peaks are getting lower and the valleys are getting deeper. There is a weak negative trend in the data. (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
GaringerOzone.monthly.components <- as.data.frame(GaringerOzone.monthly.decomposed$time.series[,1:3])

seasonal_component <- GaringerOzone.monthly.components$seasonal

GaringerOzone.monthly.deseasonalized <- GaringerOzone.monthly.ts - seasonal_component

#16

#Run MK test
Ozone_data_trend2 <- Kendall::MannKendall(GaringerOzone.monthly.deseasonalized)

# Inspect results
Ozone_data_trend2
```

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

```
summary(Ozone_data_trend2)
```

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

Answer: There is a weak trend still present without the seasonal component in a decline in mean monthly Ozone concentrations. This makes sense because even without the seasonal valleys and peaks, the overall trend is still decreasing.