Assignment 7: Time Series Analysis

Sophia Bryson

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/Assignments"
```

```
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   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

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

```
library(trend)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
theme_set(theme_minimal())
```

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

# Get list of files to pull
datasetList <- dir("..//Data//Raw//Ozone_TimeSeries")

# Pull first one outside of loop to have a seed onto which to bind
ds <- read.csv(file = paste0("..//Data//Raw//Ozone_TimeSeries//", datasetList[1]),
               stringsAsFactors = TRUE)
GaringerOzone <- ds

# Pull rest and bind to first with loop
  for (i in 2:length(datasetList)) {
    ds <- read.csv(file = paste0("..//Data//Raw//Ozone_TimeSeries//", datasetList[i]),
                   stringsAsFactors = TRUE)
    GaringerOzone <- rbind(GaringerOzone, ds)
  }

# Check it out
summary(GaringerOzone); dim(GaringerOzone)
```

```
##          Date         Source        Site.ID               POC
##  01/01/2010:   1   AQS   :3588   Min.   :371190041   Min.   :1
##  01/02/2010:   1   AirNow:   1   1st Qu.:371190041   1st Qu.:1
##  01/03/2010:   1                 Median :371190041   Median :1
##  01/04/2010:   1                 Mean   :371190041   Mean   :1
##  01/05/2010:   1                 3rd Qu.:371190041   3rd Qu.:1
##  01/07/2010:   1                 Max.   :371190041   Max.   :1
##  (Other)   :3583
##  Daily.Max.8.hour.Ozone.Concentration UNITS      DAILY_AQI_VALUE
##  Min.   :0.00200                      ppm:3589   Min.   :  2.00
##  1st Qu.:0.03200                                 1st Qu.: 30.00
##  Median :0.04100                                 Median : 38.00
##  Mean   :0.04163                                 Mean   : 41.57
##  3rd Qu.:0.05100                                 3rd Qu.: 47.00
##  Max.   :0.09300                                 Max.   :169.00
##
##                  Site.Name    DAILY_OBS_COUNT PERCENT_COMPLETE
##  Garinger High School:3589   Min.   : 6.00   Min.   : 35.0
##                              1st Qu.:17.00   1st Qu.:100.0
##                                Median :17.00   Median :100.0
##                                Mean   :16.97   Mean   : 99.8
##                                3rd Qu.:17.00   3rd Qu.:100.0
##                                Max.   :19.00   Max.   :100.0
##
##  AQS_PARAMETER_CODE AQS_PARAMETER_DESC   CBSA_CODE
##  Min.   :44201      Ozone:3589         Min.   :16740
##  1st Qu.:44201                         1st Qu.:16740
##  Median :44201                         Median :16740
##  Mean   :44201                         Mean   :16740
##  3rd Qu.:44201                         3rd Qu.:16740
##  Max.   :44201                         Max.   :16740
##
##                                CBSA_NAME      STATE_CODE            STATE
##  Charlotte-Concord-Gastonia, NC-SC:3589   Min.   :37   North Carolina:3589
##                                           1st Qu.:37
##                                           Median :37
##                                           Mean   :37
##                                           3rd Qu.:37
##                                           Max.   :37
##
##   COUNTY_CODE          COUNTY     SITE_LATITUDE   SITE_LONGITUDE
##  Min.   :119   Mecklenburg:3589   Min.   :35.24   Min.   :-80.79
##  1st Qu.:119                      1st Qu.:35.24   1st Qu.:-80.79
##  Median :119                      Median :35.24   Median :-80.79
##  Mean   :119                      Mean   :35.24   Mean   :-80.79
##  3rd Qu.:119                      3rd Qu.:35.24   3rd Qu.:-80.79
##  Max.   :119                      Max.   :35.24   Max.   :-80.79
##
```

```
## [1] 3589   20
```

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.

```r
# 3
GaringerOzone <- GaringerOzone %>% mutate(Date = as.Date(Date, format = "%m/%d/%Y"))

# 4
GaringerOzone <- GaringerOzone %>% select(Date, Daily.Max.8.hour.Ozone.Concentration,
                                          DAILY_AQI_VALUE)

# 5
Days <- as.data.frame(seq(from = ymd("2010-01-01"),
                          to = ymd("2019-12-31"),
                          by = "day"))
names(Days) <- "Date"

# 6
GaringerOzone <- left_join(Days, GaringerOzone, by = "Date")
```

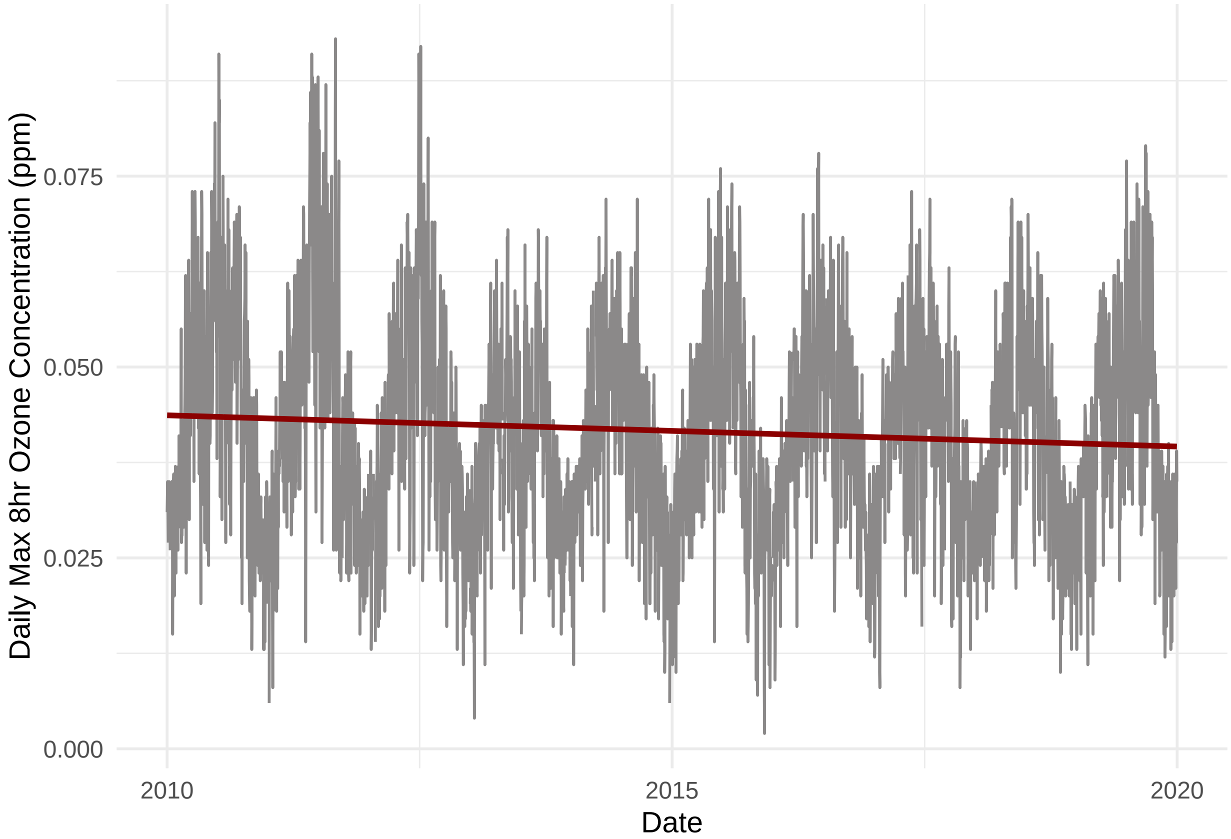
Visualize

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
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line(color = "snow4") +
  geom_smooth(method = "lm", se = FALSE, color = "red4") +
  labs(y = "Daily Max 8hr Ozone Concentration (ppm)")
```

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

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



Answer: The plot suggests a slight downward trend (reduction in max daily 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 piece wise constant or spline interpolation?

```
#8
GaringerOzone.daily <- GaringerOzone %>% mutate(DailyOzone = na.approx(Daily.Max.8.hour.Ozone.Concentrat
                                                DailyAQI = na.approx(DAILY_AQI_VALUE)) %>%
                                         select(Date, DailyOzone, DailyAQI)
```

> Answer: A linear interpolation is a more appropriate choice than a constant or spline interpolation due to the fluctutations in the data (oscillations upward and downward). A linear interpolation will use the points on either side of missing data to generate a value, while a piecewise constant approach would assign the value of the nearest neighbor (skewing the interpolated values too high or too low), and a spline interpolation would use a higher-order function, which would overcomplicate the interpolation beyond what is neeeded in thise case.

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)

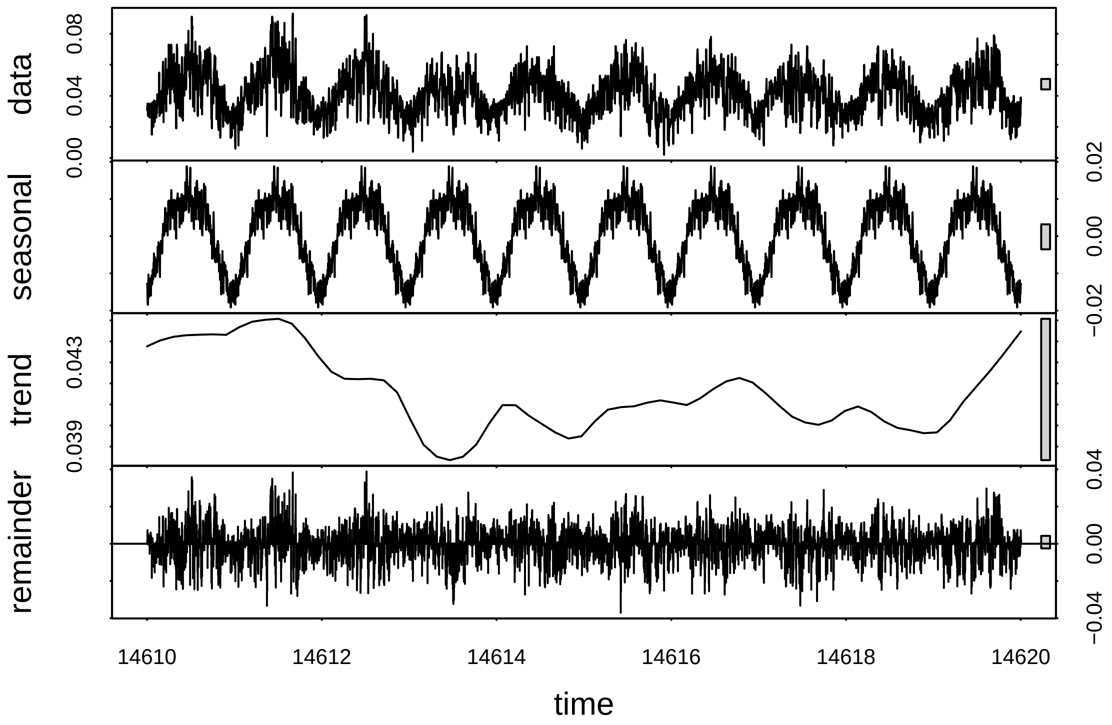
```
#9
GaringerOzone.monthly <- GaringerOzone.daily %>%
                         mutate(Month = month(Date),
                                Year = year(Date)) %>%
                         group_by(Year, Month) %>%
                         summarise(MeanOzone = mean(DailyOzone), .groups = "drop") %>%
                         mutate(DispDate = as.Date(paste0(Year, "-", 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.

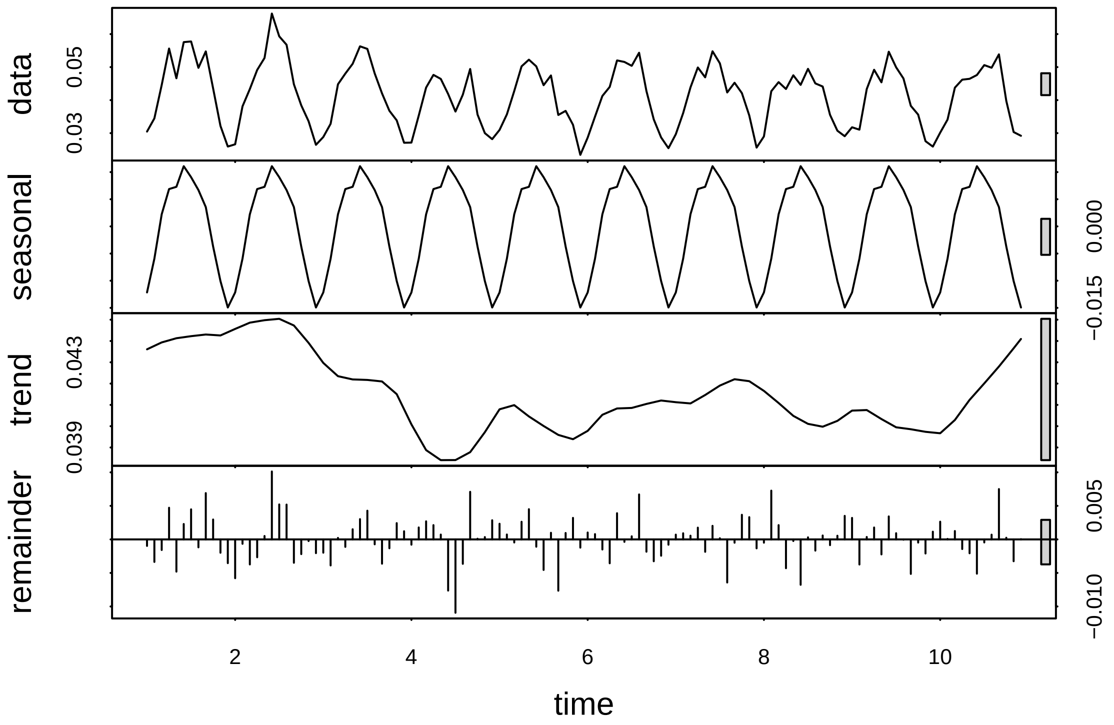
```
#10
GaringerOzone.daily.ts <- GaringerOzone.daily$DailyOzone %>%
                          ts(start = min(GaringerOzone.daily$Date), frequency = 365)

GaringerOzone.monthly.ts <- GaringerOzone.monthly$MeanOzone %>%
                            ts(start = min(c(GaringerOzone.monthly$Year,
                                             GaringerOzone.monthly$Month)),
                               frequency = 12)
```

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

```
#11
DailyDecomp <- GaringerOzone.daily.ts %>% stl(s.window = "periodic")
plot(DailyDecomp)
```



```
MonthlyDecomp <- GaringerOzone.monthly.ts %>% stl(s.window = "periodic")
plot(MonthlyDecomp)
```



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
monthly_ozone_trend <- smk.test(GaringerOzone.monthly.ts)

# Inspect results
monthly_ozone_trend
```

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

```
summary(monthly_ozone_trend)
```

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

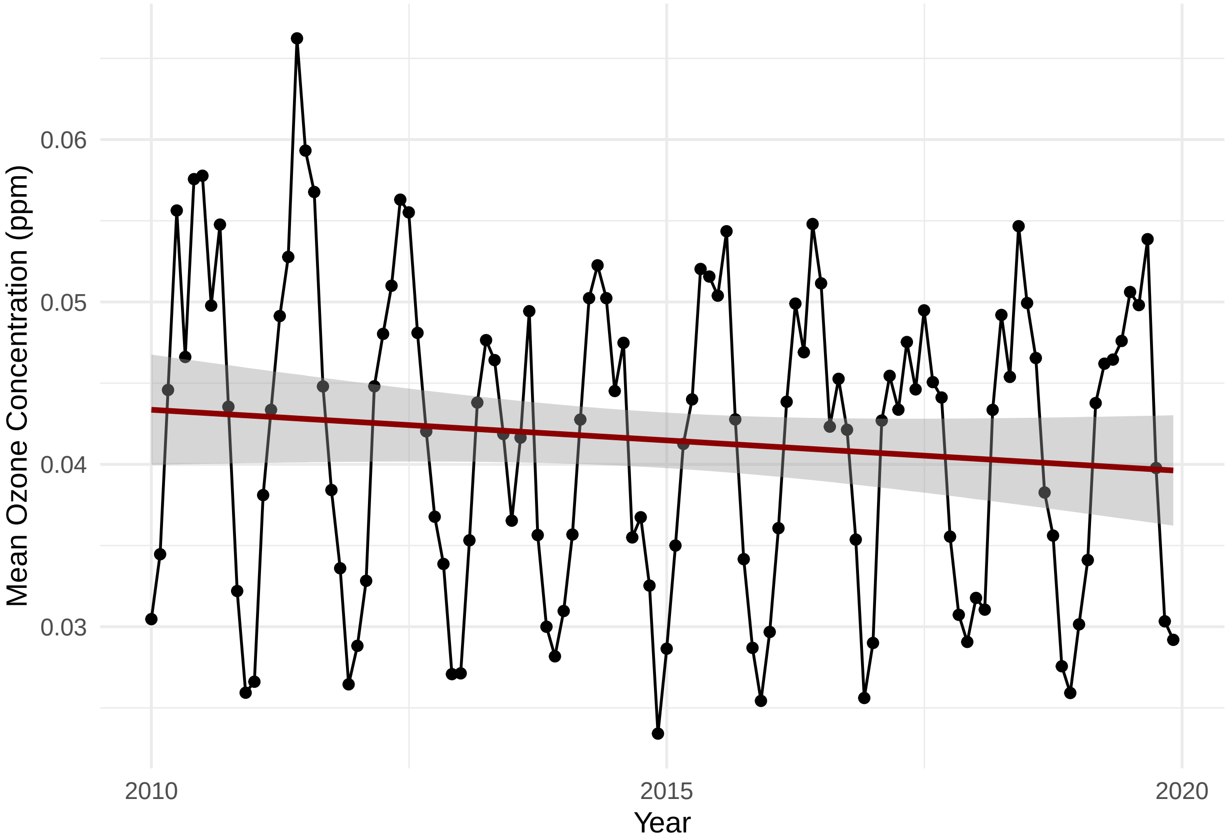
```
## Statistics for individual seasons
##
## H0
##                      S  varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15   125  0.333  1.252  0.21050
## Season 2:   S = 0   -1   125 -0.022  0.000  1.00000
## Season 3:   S = 0   -4   124 -0.090 -0.269  0.78762
## Season 4:   S = 0  -17   125 -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 the plotting of the decomposed time series above shows a strong seasonal signal in the ozone concentration data, and the seasonal Mann Kendall is best suited to appropriately accounting for this cycle.

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

#Visualization
ozone_data_plot <- ggplot(GaringerOzone.monthly, aes(x = DispDate, y = MeanOzone)) +
                   geom_point() +
                   geom_line() +
                   labs(x = "Year",
                        y = "Mean Ozone Concentration (ppm)") +
                   geom_smooth( method = lm, color = "red4")
ozone_data_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: Over the 2010s, there has been a slight decrease in ozone concentration at the Garinger station. The magnitude of the decline is substantially less than the intra-annual seasonal variation in ozone concentration, but nonetheless significant (p = 0.04965).

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.

```r
#15
GaringerOzone.monthly_Components <- as.data.frame(MonthlyDecomp$time.series)
GaringerOzone.monthly_nonSeasonal <- GaringerOzone.monthly_Components %>%
                                     select(-seasonal)

#16
nonSeasonal.ts <- ts(GaringerOzone.monthly_nonSeasonal,
                     start = min(c(GaringerOzone.monthly$Year,
                                   GaringerOzone.monthly$Month)),
                     frequency = 12)
nonSeasonalMonthly_trend <- smk.test(nonSeasonal.ts)

# Inspect results
nonSeasonalMonthly_trend
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  nonSeasonal.ts
## z = -12.41, p-value < 2.2e-16
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##     S  varS
## -1326 11400
```

```
summary(nonSeasonalMonthly_trend)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: nonSeasonal.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                       S varS    tau      z   Pr(>|z|)
## Season 1:   S = 0   -96  950 -0.505 -3.082 0.00205472  **
## Season 2:   S = 0  -108  950 -0.568 -3.472 0.00051749 ***
## Season 3:   S = 0  -100  950 -0.526 -3.212 0.00131822  **
## Season 4:   S = 0  -128  950 -0.674 -4.120 3.7818e-05 ***
## Season 5:   S = 0  -122  950 -0.642 -3.926 8.6457e-05 ***
## Season 6:   S = 0  -124  950 -0.653 -3.991 6.5893e-05 ***
## Season 7:   S = 0  -126  950 -0.663 -4.056 5.0020e-05 ***
## Season 8:   S = 0  -108  950 -0.568 -3.472 0.00051749 ***
## Season 9:   S = 0  -102  950 -0.537 -3.277 0.00104964  **
## Season 10:   S = 0 -102  950 -0.537 -3.277 0.00104964  **
## Season 11:   S = 0 -116  950 -0.611 -3.731 0.00019065 ***
## Season 12:   S = 0  -94  950 -0.495 -3.017 0.00255022  **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

> Answer: When seasonal variation is removed, the signalled decline in ozone concentrations is more pronounced ($p < 2.2e{-}16$). This mkes sense, given the magnitude of the intraannual seasonal variation in ozone concentrations.