Assignment 7: Time Series Analysis

Maggie O'Shea

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] "/Users/maggieoshea/Desktop/Spring 2022/Data Analytics/Environmental_Data_Analytics_2022/Assignme
```

```
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(zoo)
```

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

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

```
library(lubridate)
```

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

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

```
library(trend)

A7theme <- theme_gray(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
            legend.position = "left")

theme_set(A7theme)
```

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

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
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")
class(GaringerOzone$Date)
```

```
## [1] "Date"
```

```
# 4
wrangle.garingerozone <- 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)[1] <- "Date"

# 6
GaringerOzone <- left_join(Days, wrangle.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?

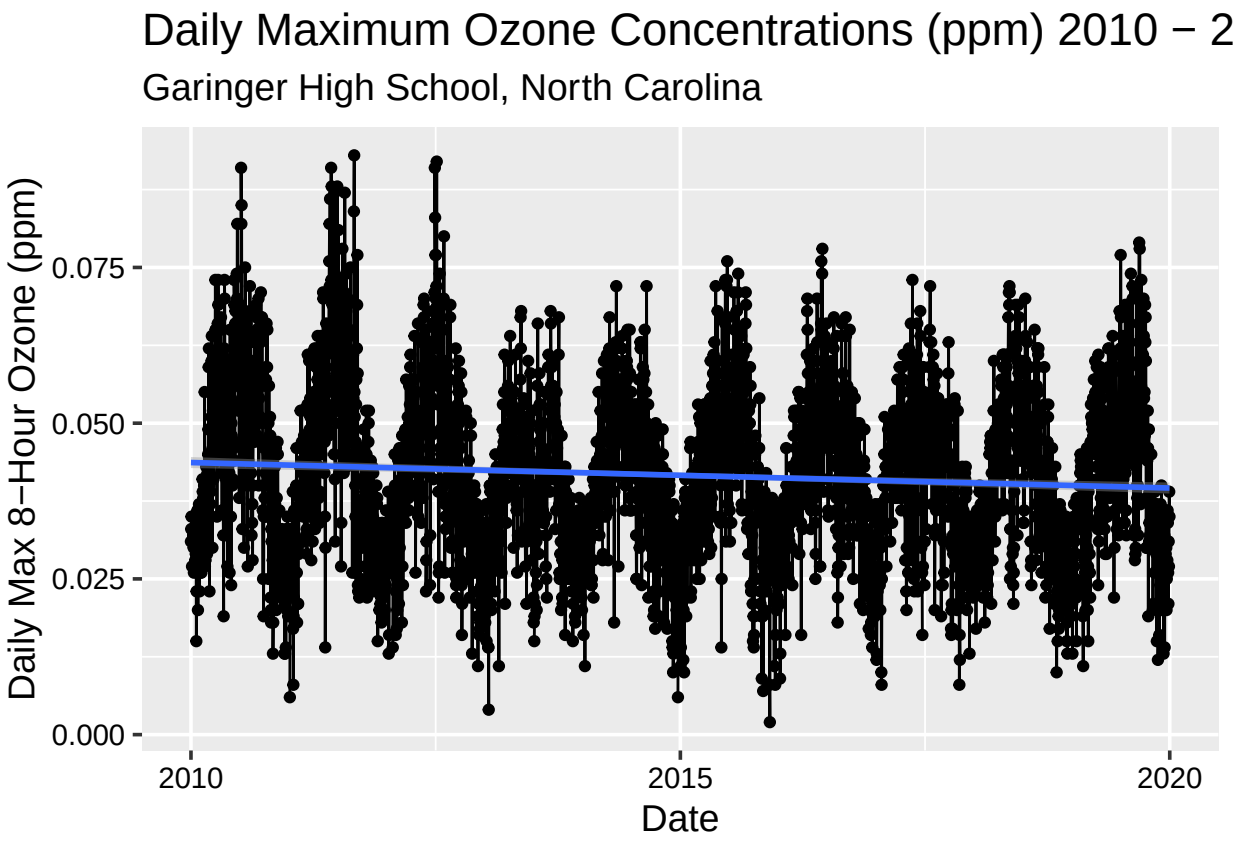
```
#7
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_point() +
  geom_line() +
```

```
  ylab("Daily Max 8-Hour Ozone (ppm)") +
  xlab("Date") +
  geom_smooth( method = lm )+
  labs(title = "Daily Maximum Ozone Concentrations (ppm) 2010 - 2019",
       subtitle= "Garinger High School, North Carolina" )
```

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

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

```
## Warning: Removed 63 rows containing missing values (geom_point).
```



Answer: This plot suggests there is a slight trend, specifically a decline 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
garingerozone_clean <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration) )
```

Answer: Ozone appears to have a seasonanality to it that could be better maintained with a linear approach. A piecewise constant approach may not capture these trends as it would simply assign the value of the nearest observation to that point which may not best represent these shifts. Similarly, the Ozone data has linear trends rather than curved lines, so the spline method may overfit the data and fill in the gap with a quadratic curve that does not match the rest of the data.

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_clean%>%
  mutate(Month=month(Date))%>%
  mutate(Year=year(Date))%>%
  group_by(Month, Year)%>%
  summarise(mean_ozone = mean(Daily.Max.8.hour.Ozone.Concentration))%>%
  mutate(Month_Year = my(paste0(Month, "_", Year)))
```

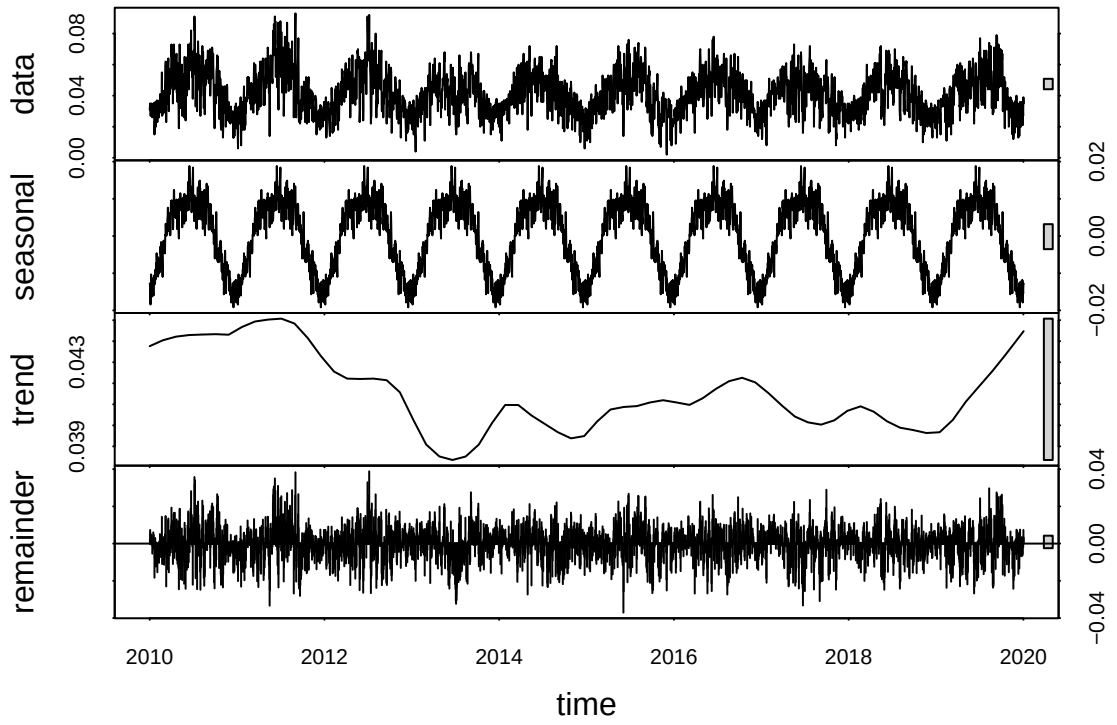
```
## `summarise()` has grouped output by 'Month'. You can override using the `.groups` argument.
```

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.

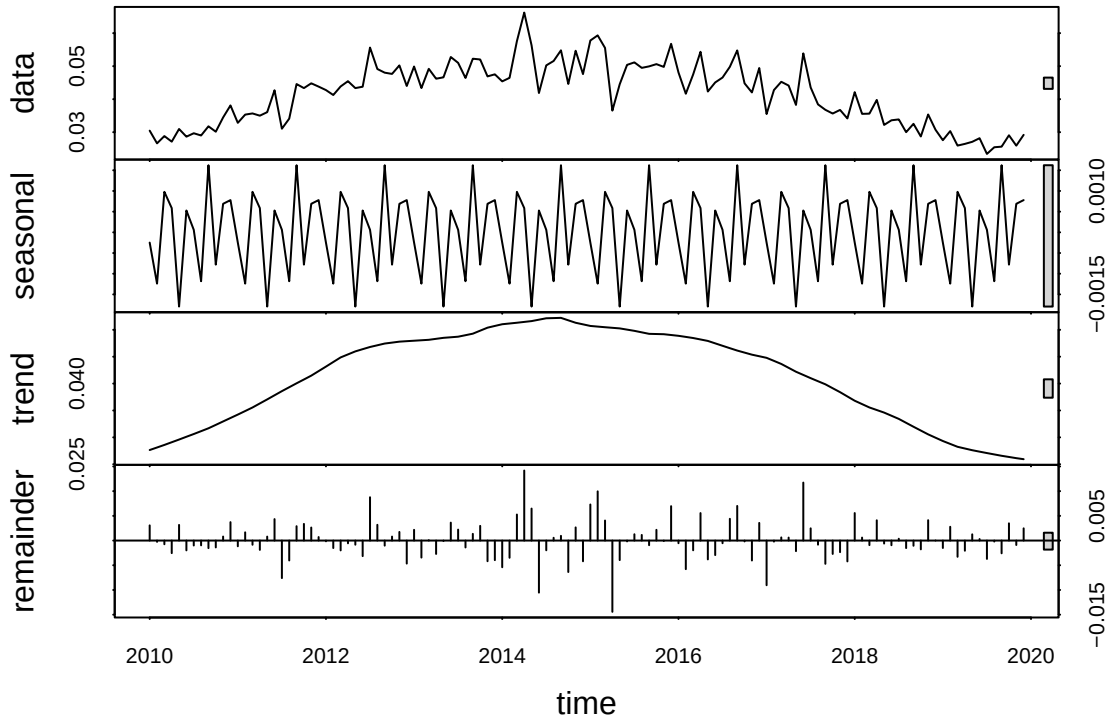
```
#10
f_month.day <- month(first(garingerozone_clean$Date))
f_year.day <- year(first(garingerozone_clean$Date))
GaringerOzone.daily.ts <- ts(garingerozone_clean$Daily.Max.8.hour.Ozone.Concentration,
                   start=c(f_year.day,f_month.day),
                   frequency=365)

f_month.month<- month(first(GaringerOzone.monthly$Month_Year))
f_year.month <-year(first(GaringerOzone.monthly$Month_Year))
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone,
                   start=c(f_year.month,f_month.month),
                   frequency=12)
```

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

```
#11
dailyozone_decomp <- stl(GaringerOzone.daily.ts,s.window = "periodic")
plot(dailyozone_decomp)
```




```
monthlyozone_decomp <- stl(GaringerOzone.monthly.ts,s.window = "periodic")
plot(monthlyozone_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
#Kendall Code
monthlyozone_smk1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Inspect results
monthlyozone_smk1
```

```
## tau = -0.1, 2-sided pvalue =0.16323
```

```
summary(monthlyozone_smk1)
```

```
## Score =  -54 , Var(Score) = 1500
## denominator =  540
## tau = -0.1, 2-sided pvalue =0.16323
```

```
#Trend Code
monthlyozone_smk2 <- trend::smk.test(GaringerOzone.monthly.ts)

#Inspect Results
monthlyozone_smk2
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.3685, p-value = 0.1712
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -54 1500
```

```
summary(monthlyozone_smk2)
```

```
##
##  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    1  125  0.022  0.000  1.00000
## Season 2:   S = 0    5  125  0.111  0.358  0.72051
## Season 3:   S = 0   -3  125 -0.067 -0.179  0.85803
## Season 4:   S = 0    1  125  0.022  0.000  1.00000
## Season 5:   S = 0   -9  125 -0.200 -0.716  0.47427
## Season 6:   S = 0    1  125  0.022  0.000  1.00000
## Season 7:   S = 0  -11  125 -0.244 -0.894  0.37109
## Season 8:   S = 0   -3  125 -0.067 -0.179  0.85803
```

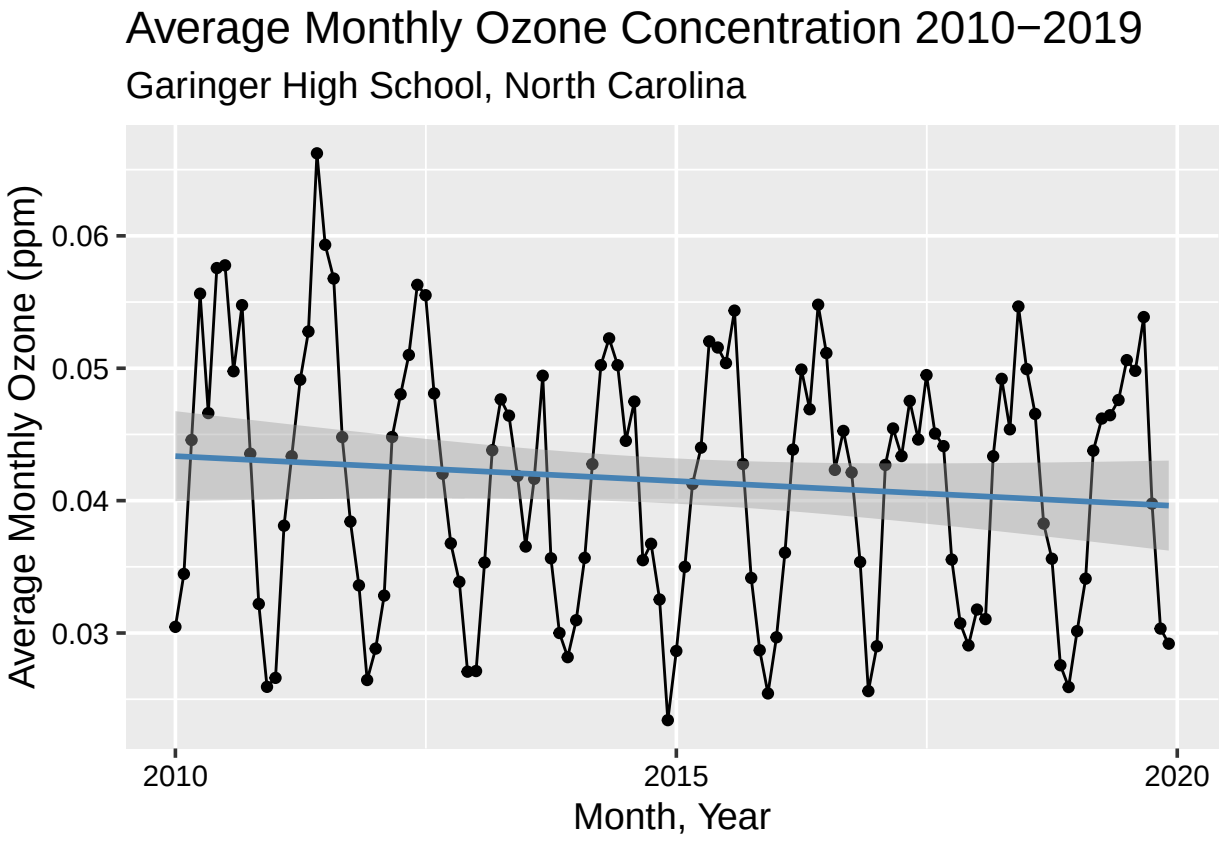
```
## Season 9:   S = 0   -5  125 -0.111 -0.358  0.72051
## Season 10:   S = 0 -11  125 -0.244 -0.894  0.37109
## Season 11:   S = 0 -15  125 -0.333 -1.252  0.21050
## Season 12:   S = 0  -5  125 -0.111 -0.358  0.72051
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: The seasonal Mann-Kendall test is most appropriate because it can test to see if a monotonic trend exists in a dataset that has a seasonal component. Because our data has a seasonal component, it is thus necessary to use this test. The other tests, in contrast, do not allow for seasonality.

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
ggplot(GaringerOzone.monthly, aes(x = Month_Year, y = mean_ozone)) +
  geom_point() +
  geom_line() +
  ylab("Average Monthly Ozone (ppm)") +
  xlab("Month, Year") +
  labs(title = "Average Monthly Ozone Concentration 2010-2019",
       subtitle= "Garinger High School, North Carolina")+
  geom_smooth( method = lm, color = "steelblue", size = 1)
```

```
## `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: The plot above suggests there is a very slight declining trend in monthly average ozone concentratoins between 2010 and 2019 at Garinger High School. However, upon running a time series analysis on this data, the analysis does not find strong evidence of this trend. Results from the seasonal Mann-Kendall test provide a p-value greater than the 0.05 significance level (p = 0.17). Given this larger p-vale, there is not evidence to reject the null hypothesis that the data is stationary. This suggests that the monthly average ozone concentration is stationary.

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

monthly_Components <- as.data.frame(monthlyozone_decomp$time.series[,1:3])

monthly_Components <- mutate(monthly_Components,
        Observed = GaringerOzone.monthly$mean_Ozone,
        Date = GaringerOzone.monthly$Month_Year)
```

```
## Warning: Unknown or uninitialised column: 'mean_Ozone'.
```

```
noseason_monthly <- mutate(monthly_Components,
        NoSeason = trend+remainder)


#16
ns_monthlyozone_ts <- ts(noseason_monthly$NoSeason,
                   start=c(f_year.month,f_month.month),
                   frequency=12)

nonseasonal.smk<-Kendall::MannKendall(ns_monthlyozone_ts)
summary(nonseasonal.smk)
```

```
## Score =  -718 , Var(Score) = 194366.7
## denominator =  7140
## tau = -0.101, 2-sided pvalue =0.10388
```

Answer: After removing the seasonal trend from the monthly ozone data, the resulting p-value was still above a 0.05 significance level, however did reduce to 0.10, and thus would meet a 0.10 significance level test.