Assignment 8: Time Series Analysis

Iddrisu Sharu Deen

Spring 2024

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

```r
library(tidyverse);library(lubridate);library(zoo);library(trend);library(here);library(knitr)
```

```
## -- Attaching core tidyverse packages ----------------------- tidyverse 2.0.0 --
## v dplyr     1.1.4      v readr     2.1.4
## v forcats   1.0.0      v stringr   1.5.0
## v ggplot2   3.4.4      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
##
## Attaching package: 'zoo'
##
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## here() starts at /home/guest/EDA_Spring2024
```

```
here()
```

```
## [1] "/home/guest/EDA_Spring2024"
```

```r
mytheme <- theme_minimal() +
  theme(
    text = element_text(size = 12),
    plot.title = element_text(face = "bold", size = 14),
    plot.subtitle = element_text(face = "italic", size = 12),
    axis.title = element_text(face = "bold"),
    axis.text = element_text(size = 8.5),
    legend.position = "bottom",
    legend.title = element_blank())

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.

```r
#1
# specifies directory containing files
data_directory <- here("Data","Raw","Ozone_TimeSeries")

# Lists all CSV files in the directory
file_list <- list.files(path = data_directory, pattern = "*.csv", full.names = TRUE)

# Import each file and combine them into a single dataframe
GaringerOzone <- file_list %>%
  lapply(read.csv) %>%
  bind_rows()

# Check dimensions of GraingerOzone
dim(GaringerOzone)
```

```
## [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$Date <- mdy(GaringerOzone$Date)
```

```
# 4
GaringerOzone <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
# 5
date_seq <- seq(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"), by = "day")
Days <- as.data.frame(date_seq)
names(Days) <- c("Date")

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

```
## Rows: 3,652
## Columns: 3
## $ Date                                 <date> 2010-01-01, 2010-01-02, 2010-01-~
## $ Daily.Max.8.hour.Ozone.Concentration <dbl> 0.031, 0.033, 0.035, 0.031, 0.027~
## $ DAILY_AQI_VALUE                      <int> 29, 31, 32, 29, 25, NA, 31, 32, 3~
```

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?

```
## $x
## [1] "Date"
##
## $y
## [1] "Daily 8-Hour Ozone Concentration (PPM)"
##
## $title
## [1] "Ozone Concentration Over Time"
##
## $caption
## [1] "Source: EPA Air Database"
##
## attr(,"class")
## [1] "labels"
```

Answer: The plot suggests a slight decreasing trend. The concentration of Daily Max 8-Hour Ozone Concentration appears to be decreasing with 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 <- GaringerOzone %>%
  mutate(ozone.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

head(GaringerOzone$ozone.clean)
```
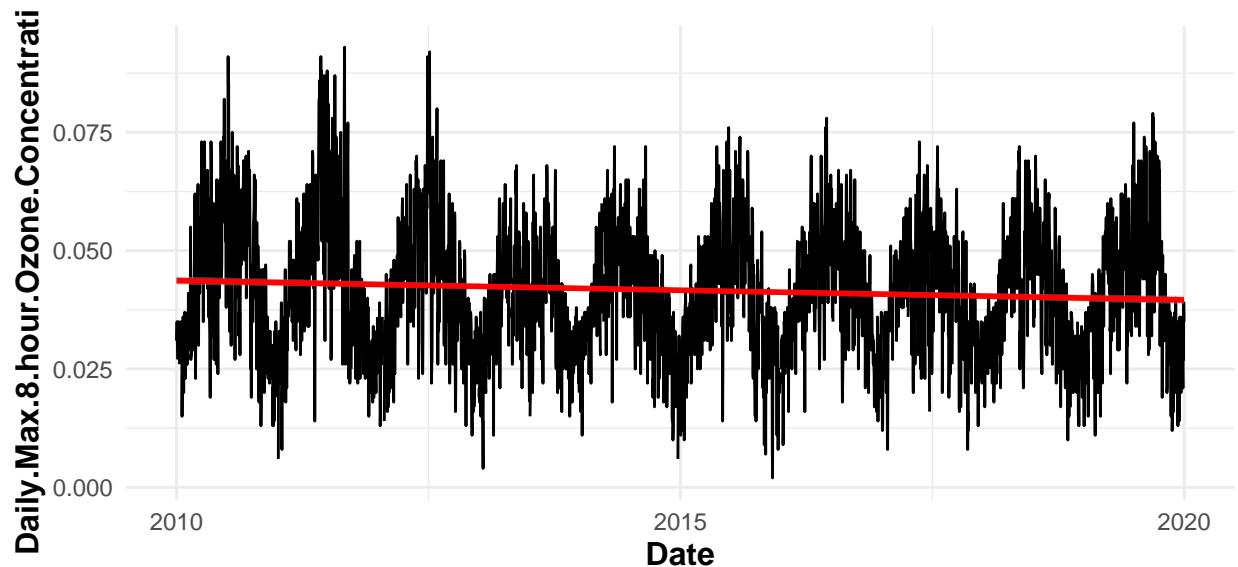
```
## [1] 0.031 0.033 0.035 0.031 0.027 0.030
```

Figure 1: Ozone Concentrations over time

Answer:

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 %>%
  mutate(Year = year(Date),
         Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarise(mean_ozone = mean(ozone.clean))
```

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

```
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(DateGroup = make_date(Year, Month, 1))
```

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 <- month(first(GaringerOzone$Date))
f_year <- year(first(GaringerOzone$Date))

GaringerOzone.daily.ts <-
  ts(GaringerOzone$ozone.clean,
     start = c(f_year, f_month),
     frequency = 365)
```

```
f_month2 <- first(GaringerOzone$Date)
f_year2 <- first(GaringerOzone$Date)

GaringerOzone.monthly.ts <-
  ts(GaringerOzone.monthly$mean_ozone,
     start = c(f_year2, f_month2),
     frequency = 12)
```

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

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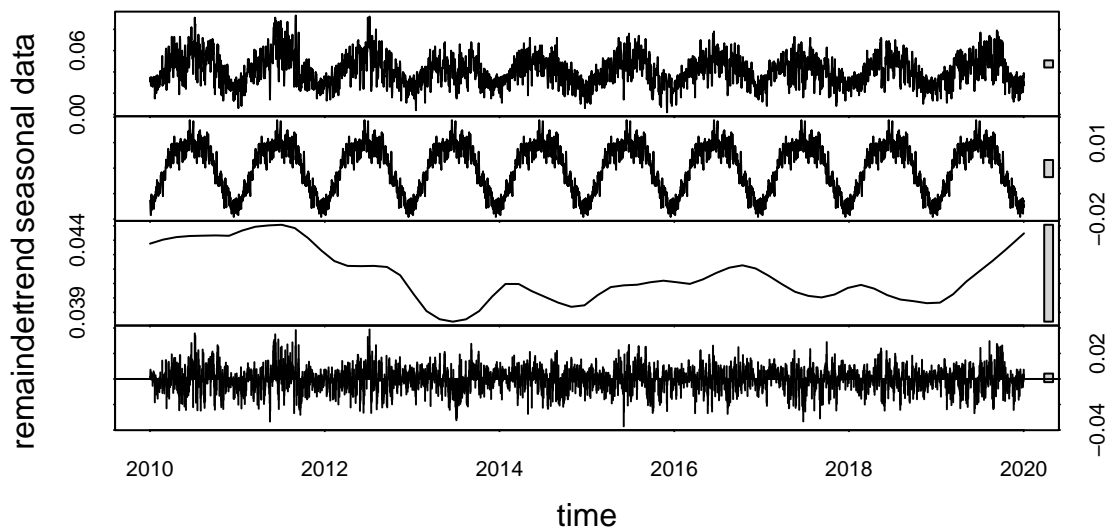

Figure 2: Decomposition

```
GaringerOzone.monthly.decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(GaringerOzone.monthly.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
smk.test(GaringerOzone.monthly.ts)

##
##  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:
```
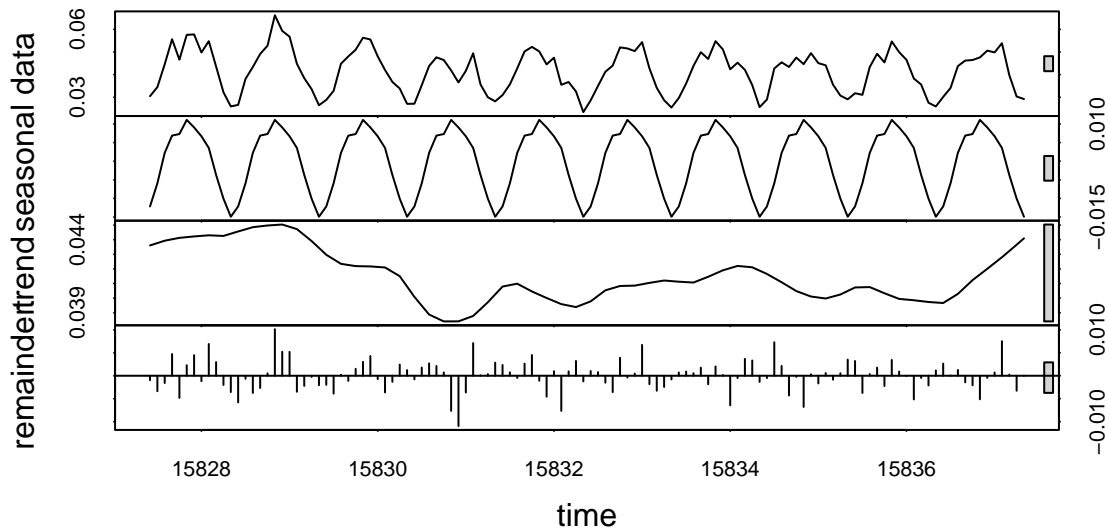
Figure 3: Decomposition

```
##     S varS
## -77 1499
```

Answer: The data exhibits some seasonaility with a significant p-value of less than 0.05.

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.plt2 <- ggplot(GaringerOzone.monthly, aes(x=DateGroup, y=mean_ozone)) +
  geom_point() +
  geom_line() +
  labs(
    title = "Mean Ozone Concentration",
    caption = "Source: EPA Air Database",
    x = "Date",
    y = "Monthly Mean Ozone Conentration (PPM)"
    )

GaringerOzone.plt2
```

Mean Ozone Concentration
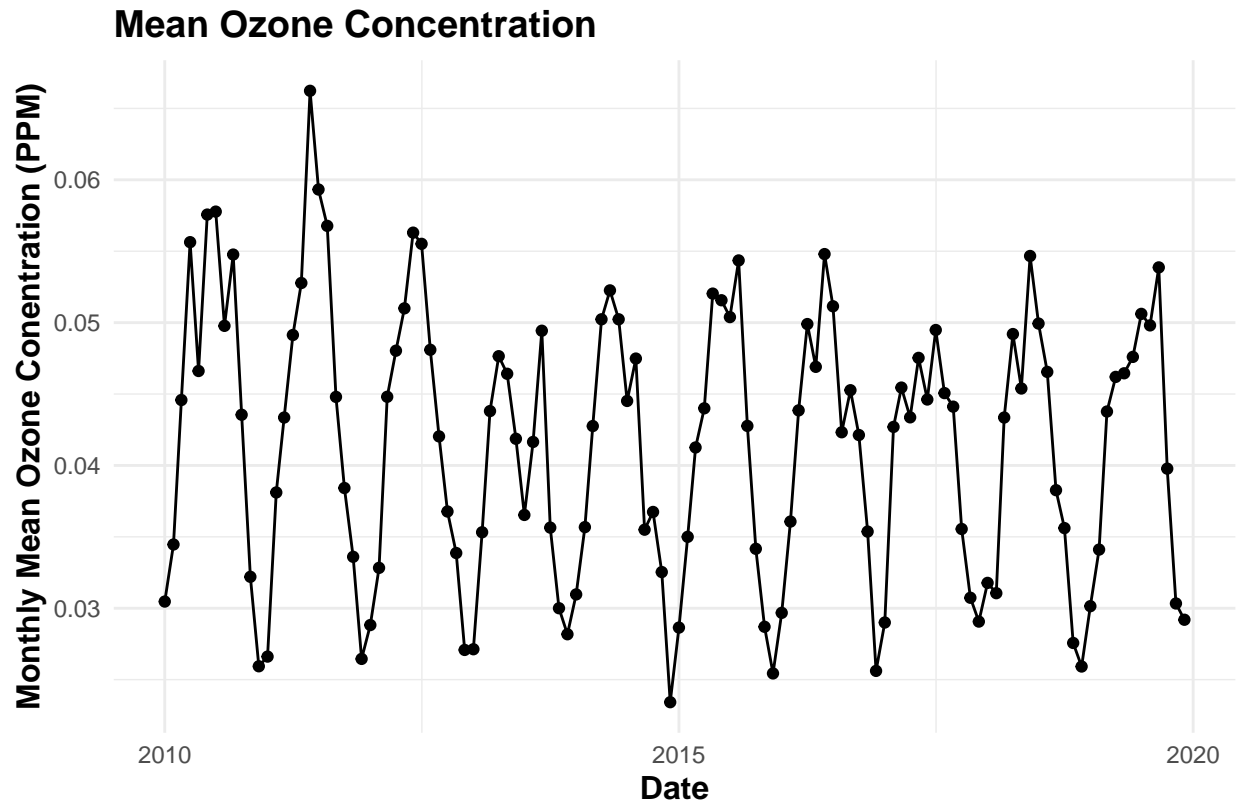
Source: EPA Air Database

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 graphic suggests a slight negative trend in ozone concentrations over time. Nonetheless, the MannKendall test shows a small yet statistically significant monotonic negative trend over the time period.

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

Garinger.components <- Garinger.components %>%
  mutate(
    Observed = GaringerOzone.monthly$mean_ozone,
    Difference = seasonal - Observed,
    Date = GaringerOzone.monthly$DateGroup
  )

Garinger.observed.plt <- ggplot(Garinger.components) +
  geom_line(aes(y=Observed, x=Date, color = "Observed")) +
  geom_line(aes(y=trend, x=Date, color = "Trend")) +
  geom_hline(yintercept=0, lty=2) +
```

```
labs(title = "Mean Monthly Ozone Concentration Trend",
     caption = "Source: EPA Air Database",
     color = NULL,
     x = "Date",
     y = "Observed Ozone Concentration (PPM)") +
scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
scale_color_manual(values = c(Observed = "black", Trend = "red"),
                   labels = c(Observed = "Observed", Trend = "Trend"),
                   limits = c("Observed", "Trend"))
```

```
Garinger.observed.plt
```



Mean Monthly Ozone Concentration Trend
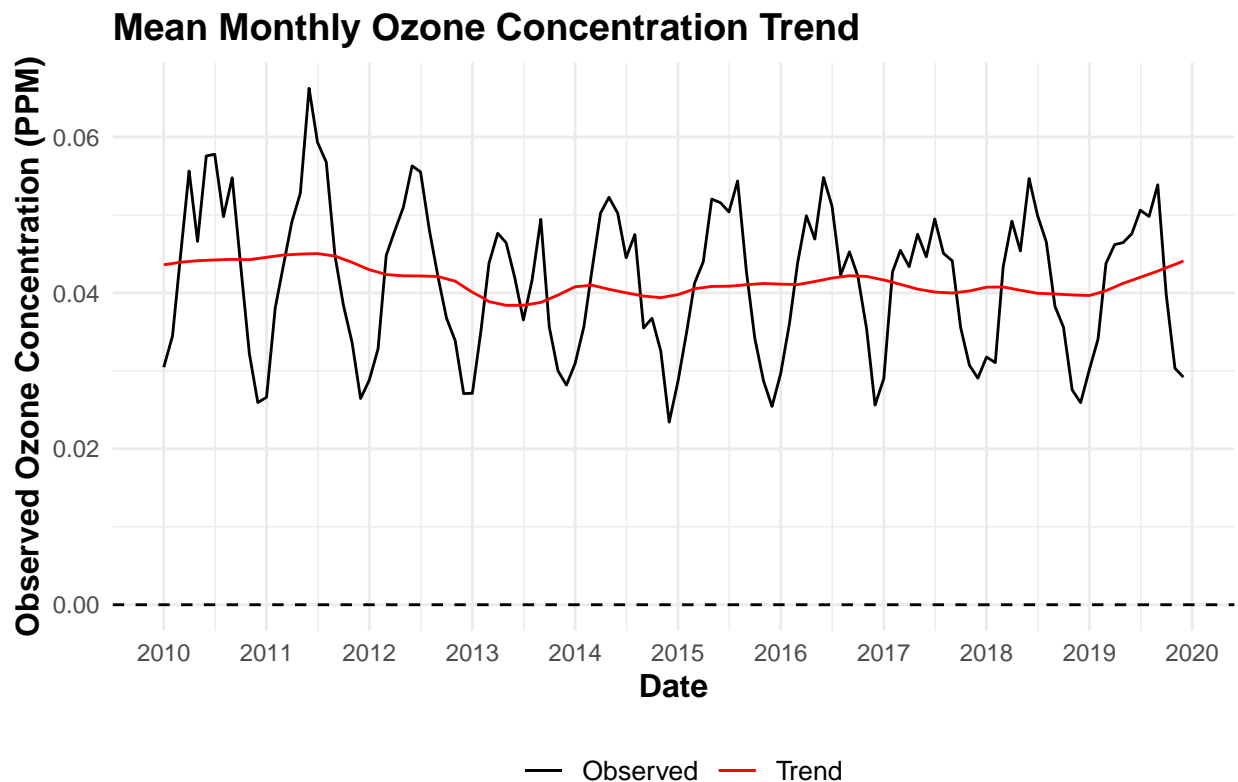
Source: EPA Air Database

```
Garinger.seasonal.plt <- ggplot(Garinger.components) +
  geom_line(aes(y=Observed, x=Date, color = "Observed")) +
  geom_line(aes(y=seasonal, x=Date, color = "Seasonal")) +
  geom_hline(yintercept=0, lty=2) +
  labs(title = "Seaosnal Mean Monthly Ozone Concentration",
       caption = "Source: EPA Air Database",
       color = NULL,
       x = "Date",
       y = "Observed Ozone Concentration (PPM)") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  scale_color_manual(values = c(Observed = "black", Seasonal = "red"),
                     labels = c(Observed = "Observed", Seasonal = "Seasonal"),
                     limits = c("Observed", "Seasonal"))
```

Seaosnal Mean Monthly Ozone Concentration
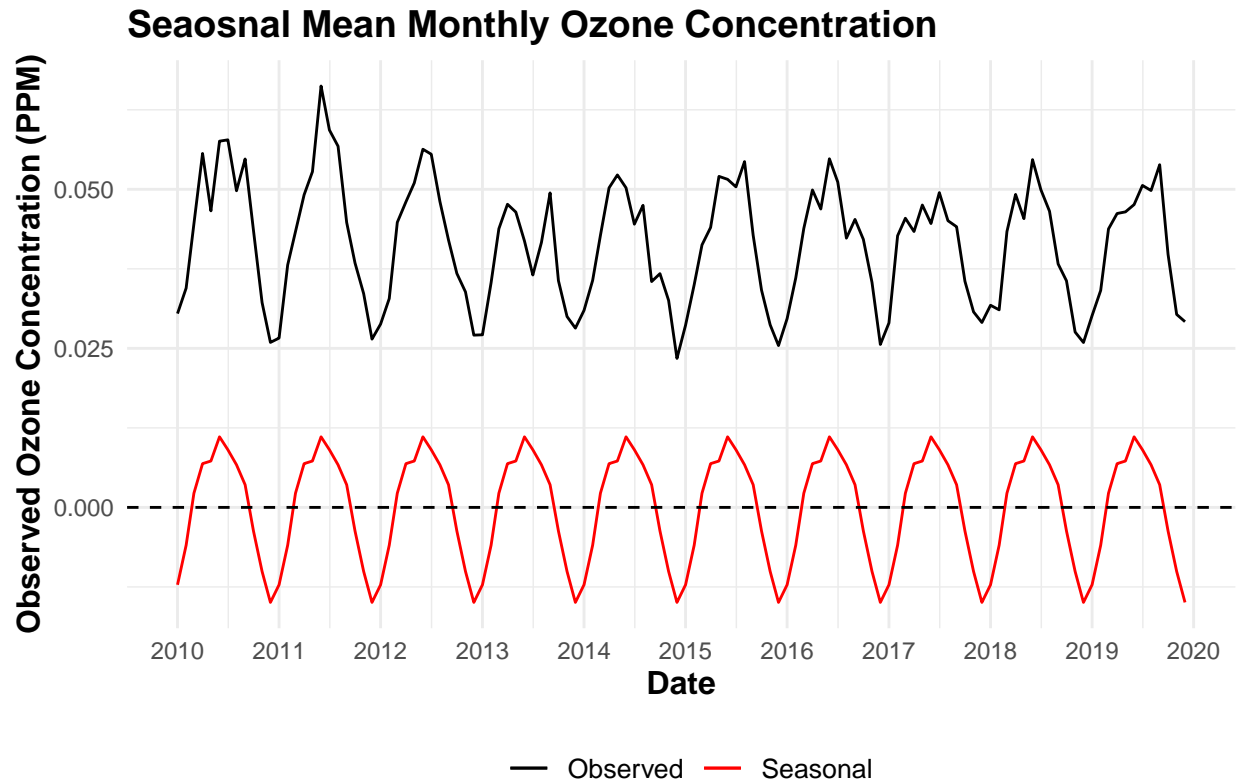
Source: EPA Air Database

```
#16
GaringerOzone.monthly.nonseasonal.ts <-
  ts(Garinger.components$Difference,
     start=c(f_year2, f_month2),
     frequency = 12)
```

```
mk.test(GaringerOzone.monthly.nonseasonal.ts) #Mann-Kendall trend test
```

```
##
##  Mann-Kendall trend test
##
## data:  GaringerOzone.monthly.nonseasonal.ts
## z = 2.672, n = 120, p-value = 0.00754
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S          varS          tau
## 1.179000e+03 1.943657e+05 1.651376e-01
```

Answer: The result suggests that the data has trend not equal to zero which is statistically significant at p-value less than 0.05.