Assignment 8: Time Series Analysis

Kayla Emerson

Fall 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

```
#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)
library(zoo)
```

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

```
library(trend)
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
```

```
#check working directory
here()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
#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.

```
#2
#import datasets
garinger10 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
                       stringsAsFactors = TRUE)
garinger11 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
                       stringsAsFactors = TRUE)
garinger12 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
                       stringsAsFactors = TRUE)
garinger13 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
                       stringsAsFactors = TRUE)
garinger14 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
                       stringsAsFactors = TRUE)
garinger15 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
                       stringsAsFactors = TRUE)
garinger16 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
                       stringsAsFactors = TRUE)
garinger17 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
                       stringsAsFactors = TRUE)
garinger18 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
                       stringsAsFactors = TRUE)
garinger19 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
                       stringsAsFactors = TRUE)

#combine datasets into one dataframe
```

```
GaringerOzone <- rbind(garinger10, garinger11, garinger12, garinger13,
                       garinger14, garinger15, garinger16, garinger17,
                       garinger18, garinger19)
```

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
#set data col as a date class
GaringerOzone$Date <- mdy(GaringerOzone$Date)
class(GaringerOzone$Date)
```

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

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

# 5
#create new data frame containing a series of dates
Days <- as.data.frame(seq(as.Date('2010-01-01'), as.Date('2019-12-31'), by = 'days'))

Days <- Days %>%
  rename(Date = `seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "days")`)

# 6
#combine data frames
GaringerOzone_processed <- Days %>%
  left_join(GaringerOzone_processed, by=c('Date'))
```

Visualize

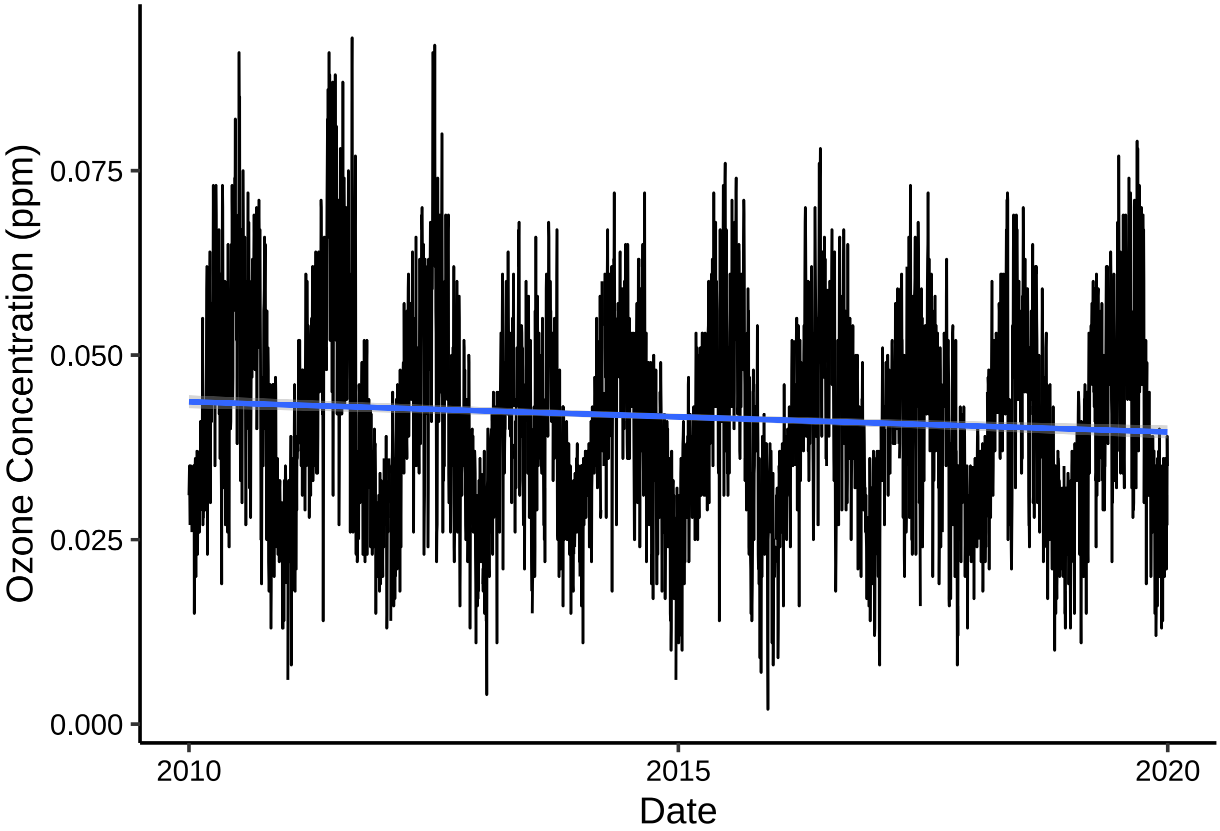
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7
#create plot depicting ozone concentrations over time
plot_one <- GaringerOzone_processed %>%
  ggplot(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = lm) +
  labs(y= "Ozone Concentration (ppm)", x  = "Date")

print(plot_one)
```

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

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



Answer:The plots suggests a slight negative trend in ozone concentration over time.There also appears to be a seasonal trend in ozone concentration.

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?

```r
#8
#linear interpolation to fill in missing data
GaringerOzone_clean <-
  GaringerOzone_processed %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration_clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentratio
```

Answer: We didn’t use a piecewise constant or spline interpolation because we are analyzing a linear regression over our data, so it makes sense to use a linear interpolation that connects dots.

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)

```r
#9
#create new data frame that contains aggregated data
GaringerOzone.monthly <- GaringerOzone_clean %>%
  mutate(year = year(Date),
         month = month(Date)) %>%
  group_by(year, month) %>%
  summarize(mean_ozone = mean(Daily.Max.8.hour.Ozone.Concentration_clean))
```

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

```r
#create a new date column with the first of each month
GaringerOzone.monthly_1 <- GaringerOzone.monthly %>%
  mutate(Date = as.Date(paste(year, month, "1", sep = "-")))
#had to make a new data frame for some reason
```

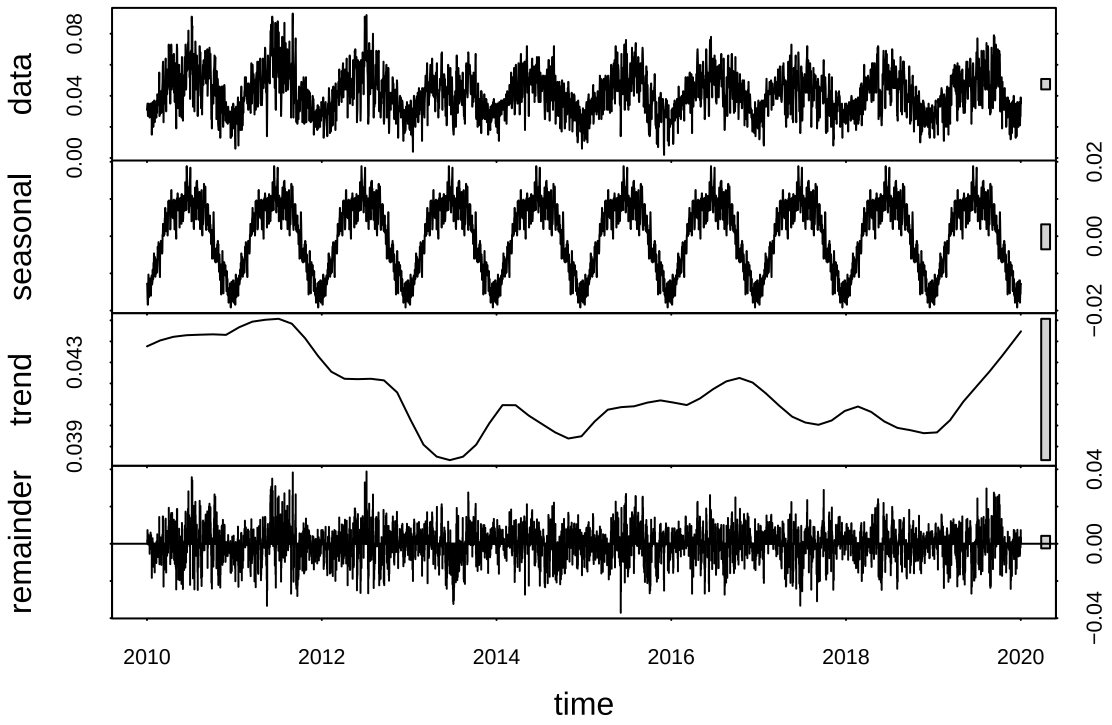
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

```r
#10
#create time series object for daily observations of ozone
GaringerOzone.daily.ts <- ts(GaringerOzone_clean$Daily.Max.8.hour.Ozone.Concentration_clean,
                             start = c(2010, 1, 1), frequency = 365)
#create time series object for monthly observations of ozone
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly_1$mean_ozone,
                             start = c(2010, 1, 1), frequency = 12)
```

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

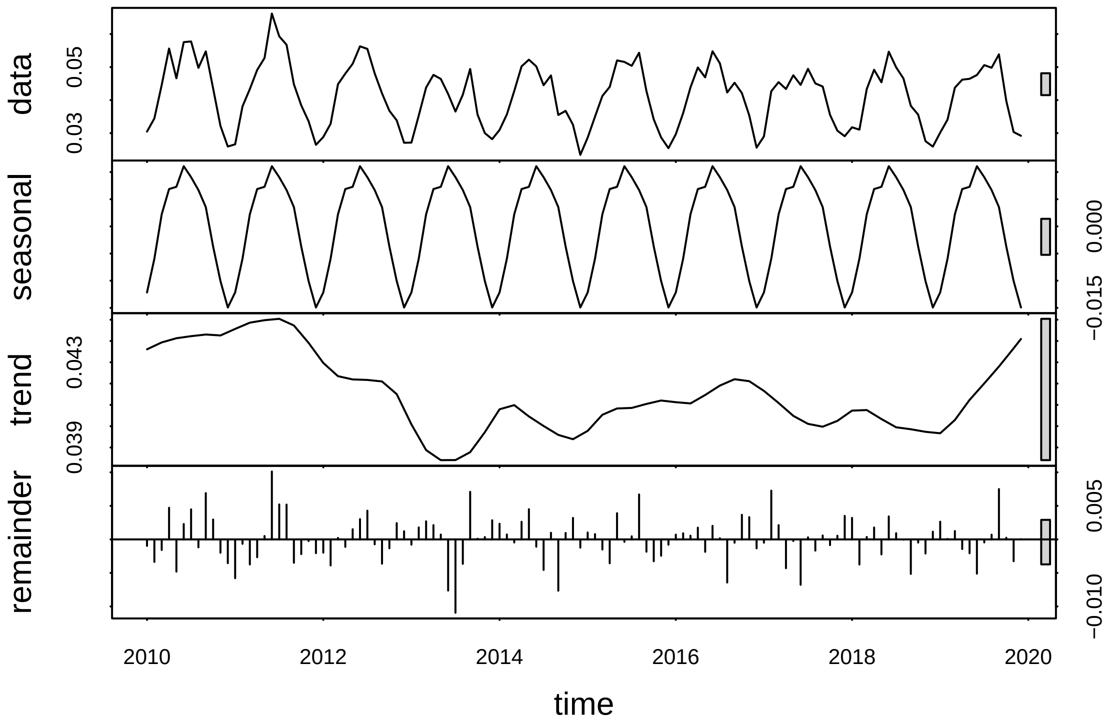
```r
#11
#decompose the daily time series object
GaringerOzone_daily_Decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")

#plot components
plot(GaringerOzone_daily_Decomposed)
```



```
#decompose the monthly time series object
GaringerOzone_monthly_Decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")

#plot components
plot(GaringerOzone_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
#run monotonic trend analysis for monthly ozone series

#mann kendall
monthly_data_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Inspect results
monthly_data_trend1
```

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

```
summary(monthly_data_trend1)
```

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

Answer: The seasonal Mann-Kendall monotonic trend analysis is the most appropriate because it takes into account seasonal variation, and ozone levels change with the seasons and we are analyzing our data on a monthly level with the assumption we will see a seasonal trend.

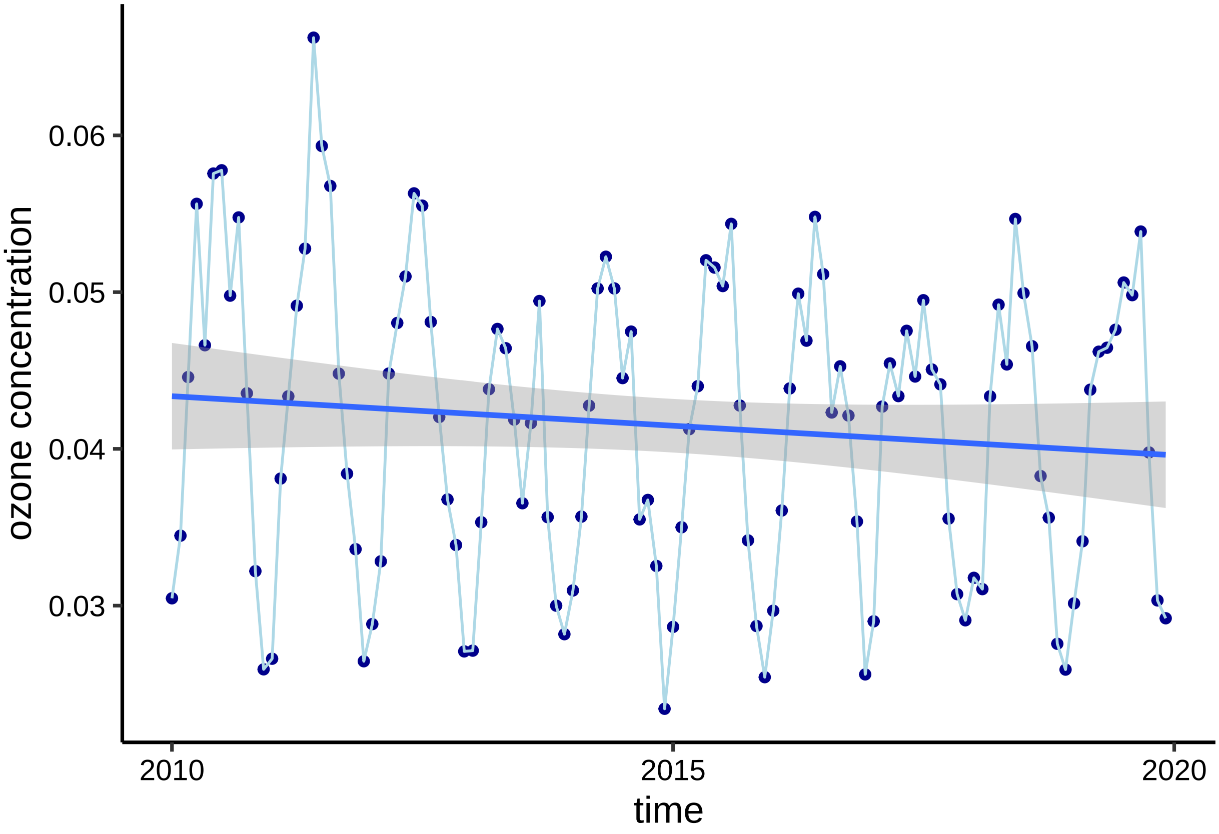
13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. Edit your axis labels accordingly.

```
# 13
#create a plot depicting mean monthly ozone concentrations over time

monthly_ozone_plot <-
ggplot(GaringerOzone.monthly_1, aes(x = Date, y = mean_ozone)) +
  geom_point(color = "darkblue") +
  geom_line(color = "lightblue") +
  ylab("ozone concentration") +
  xlab("time") +
  geom_smooth(method = lm) +
  mytheme


print(monthly_ozone_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 graph, there is a seasonal variation in ozone because the concentration changes monthly, but from year-to-year it is somewhat consistent, but it may be decreasing over time. From our seasonal mann kendall test, the p-value was 0.046724, which is less than 0.05. Therefore, we can reject the null hypothesis that there is no difference in ozone concentration between seasons.

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
#subtract the seasonal component from the monthly time series
monthly_ozone_nonseas_ts <- GaringerOzone.monthly.ts - GaringerOzone_monthly_Decomposed$time.series[,1]

#16

#run the mann kendall test
monthly_data_trend16 <- Kendall::MannKendall(monthly_ozone_nonseas_ts)

# Inspect results
monthly_data_trend16
```

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

```
summary(monthly_data_trend16)
```

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

Answer: The results of the MannKendall Seasonal test and MannKendall test are different with the former resulting in a p-value of 0.046724 and the latter resulting in a p-value of 0.0075402. However, both tests have statistically significant results.