Assignment 8: Time Series Analysis

Ashton Cloer

Fall 2023

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

```
library(here)
```

```
## here() starts at /Users/ashtoncloer
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     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
```

```r
library(lubridate)
library(zoo)
```

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

```r
library(trend)

getwd()
```

```
## [1] "/Users/ashtoncloer"
```

```r
here()
```

```
## [1] "/Users/ashtoncloer"
```

```r
mytheme1 <- theme_gray(
  base_size = 14,
  base_family = "Times") +
  theme(
    axis.text = element_text(
      color = "gray"),
        legend.position = "bottom")
theme_set(mytheme1)
```

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
GaringerNC2010.df <-
  read.csv(
    here(
    "EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
    stringsAsFactors = TRUE)

GaringerNC2011.df <-
  read.csv(
    here(
    "EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
    stringsAsFactors = TRUE)

GaringerNC2012.df <-
  read.csv(
    here(
```

```
    "EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
    stringsAsFactors = TRUE)

GaringerNC2013.df <-
  read.csv(
    here(
    "EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
    stringsAsFactors = TRUE)

GaringerNC2014.df <-
  read.csv(
    here(
    "EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
    stringsAsFactors = TRUE)

GaringerNC2015.df <-
  read.csv(
    here(
    "EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
    stringsAsFactors = TRUE)

GaringerNC2016.df <-
  read.csv(
    here(
    "EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
    stringsAsFactors = TRUE)

GaringerNC2017.df <-
  read.csv(
    here(
    "EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
    stringsAsFactors = TRUE)

GaringerNC2018.df <-
  read.csv(
    here(
    "EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
    stringsAsFactors = TRUE)

GaringerNC2019.df <-
  read.csv(
    here(
    "EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
    stringsAsFactors = TRUE)

GaringerOzone.df <- rbind(
  GaringerNC2019.df, GaringerNC2018.df,
  GaringerNC2017.df, GaringerNC2016.df,
  GaringerNC2015.df, GaringerNC2014.df,
  GaringerNC2013.df, GaringerNC2012.df,
  GaringerNC2011.df, GaringerNC2010.df)
```

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.df$Date <- as.Date(
  GaringerOzone.df$Date, format = '%m/%d/%Y')
# 4
GaringerOzone_select.df <- select(
  GaringerOzone.df, Date,
  Daily.Max.8.hour.Ozone.Concentration,
  DAILY_AQI_VALUE)
# 5
Day.df <- as.data.frame(seq(
  as.Date("2010/01/01"),
  as.Date("2019/12/31"), by = "1 day"))
colnames(Day.df)[1] = "Date"

#6
GaringerOzone.df <- left_join(
  Day.df, GaringerOzone_select.df)
```

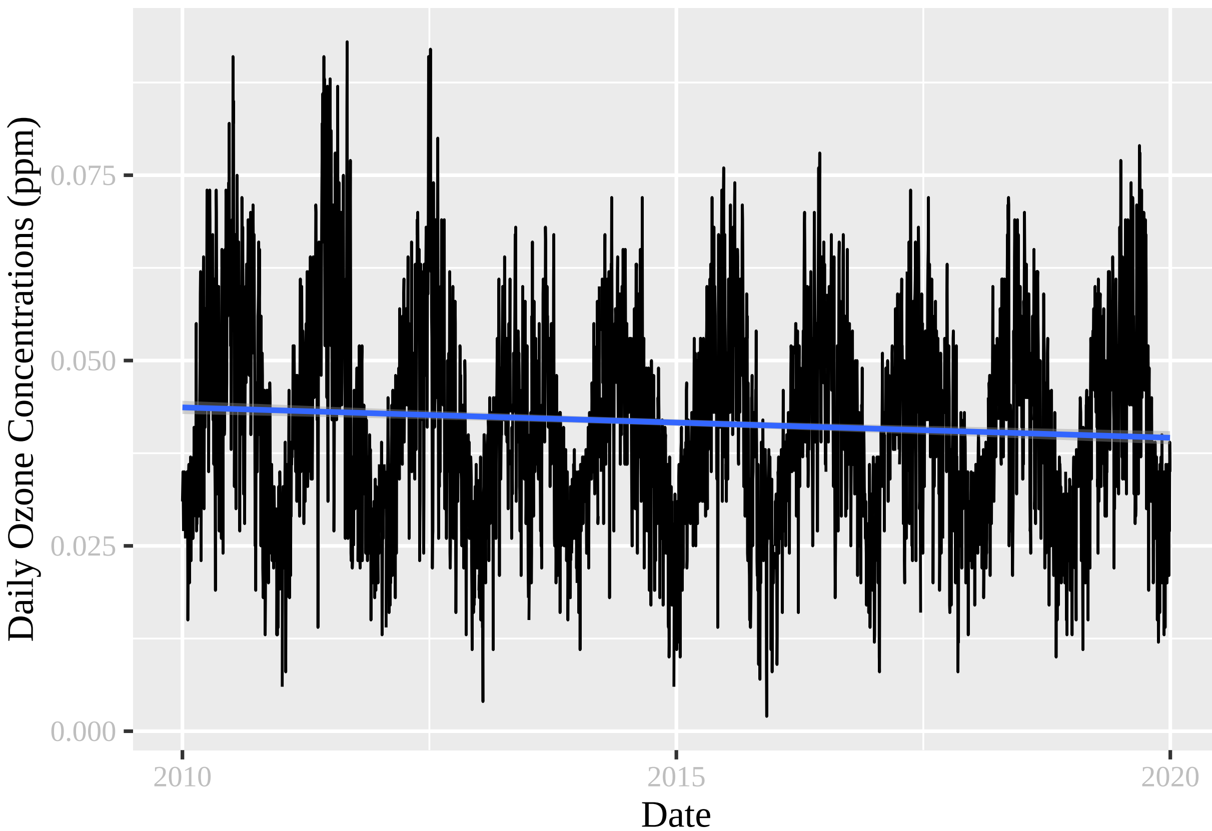
```
## Joining with `by = join_by(Date)`
```

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
ozone_bytime <- ggplot(
  GaringerOzone.df, aes(x = Date,
                        y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = "lm") +
  labs(x = "Date",
       y = "Daily Ozone Concentrations (ppm)") +
  mytheme1

print(ozone_bytime)
```

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

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



> Answer: The plot does not suggest an increasing or decreasing trend but does suggest seasonality as the data fluxuates upwards and downwards multiple times within the given time frame.

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.df <-
  GaringerOzone.df %>%
  mutate(
    Daily.Max.8.hour.Ozone.Concentration.clean = zoo::na.approx(
      Daily.Max.8.hour.Ozone.Concentration))
```

> Answer: The linear interpolation function connects dots with shortest line possible and does not change characteristics of original dataset. This is the simplest way of replacing NAs without changing the summary stasticis and any other characteristics. Compared to the linear interpolation, the spline function is best used on parametric function which this data does not contain.

And the peicewise constant would more than likely change some characteristics of the data due to its method of assuming NAs are equal to the closest point.

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.df <- GaringerOzone_clean.df %>%
  mutate(
    year = lubridate::year(Date),
    month = lubridate::month(Date)) %>%
  group_by(year, month) %>%
  summarise(
    mean.ozone.concentrations = mean(
      Daily.Max.8.hour.Ozone.Concentration.clean))
```

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

```
GaringerOzone_monthly_complete.df <- GaringerOzone_monthly.df %>%
  mutate(
    Date = my(paste0(month,"-",year))) %>%
  select(
    Date, mean.ozone.concentrations)
```

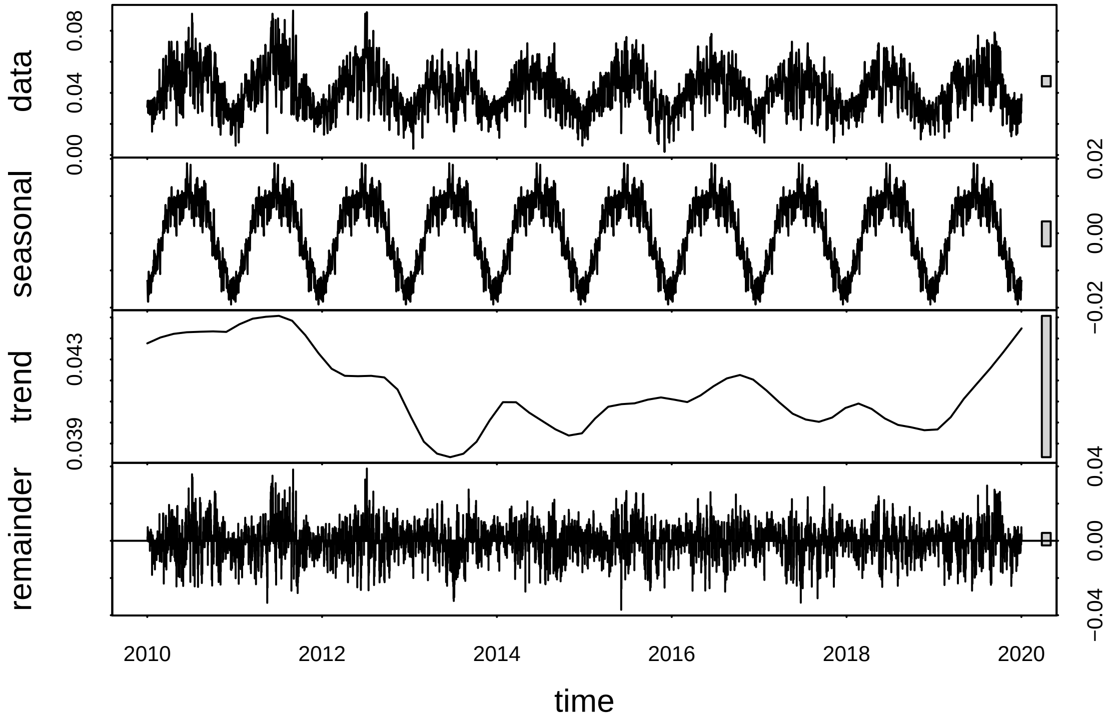
```
## Adding missing grouping variables: `year`
```

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.

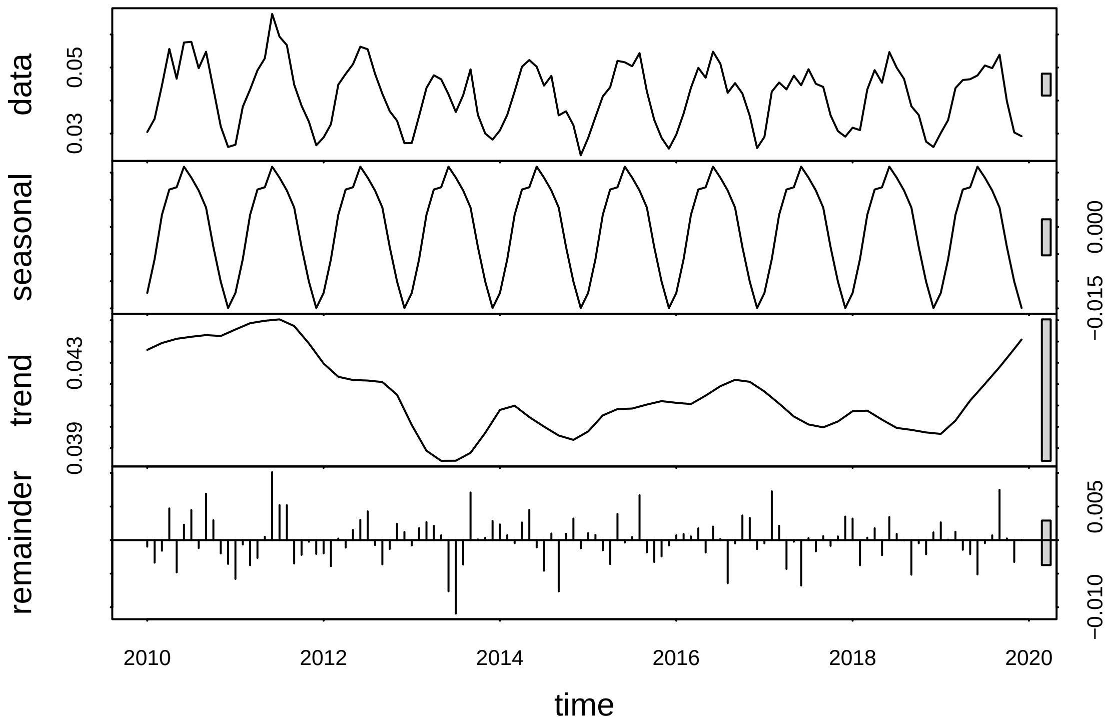
```
#10
f_month <- month(
  first(
    GaringerOzone_monthly_complete.df$Date))

f_year <- year(
  first(GaringerOzone_monthly_complete.df$Date))

GaringerOzone.daily.ts <- ts(
  GaringerOzone_clean.df$Daily.Max.8.hour.Ozone.Concentration.clean,
                   start=c(
                     f_year,f_month),
                   frequency=365)

GaringerOzone.monthly.ts <- ts(
  GaringerOzone_monthly_complete.df$mean.ozone.concentrations,
                   start=c(f_year,f_month),
                   frequency=12)
```

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

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




```
Ozone_monthly_decomp <- stl(
  GaringerOzone.monthly.ts,s.window = "periodic")
plot(Ozone_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
Seasonal_mann <- Kendall::SeasonalMannKendall(
  GaringerOzone.monthly.ts)

print(Seasonal_mann)
```

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

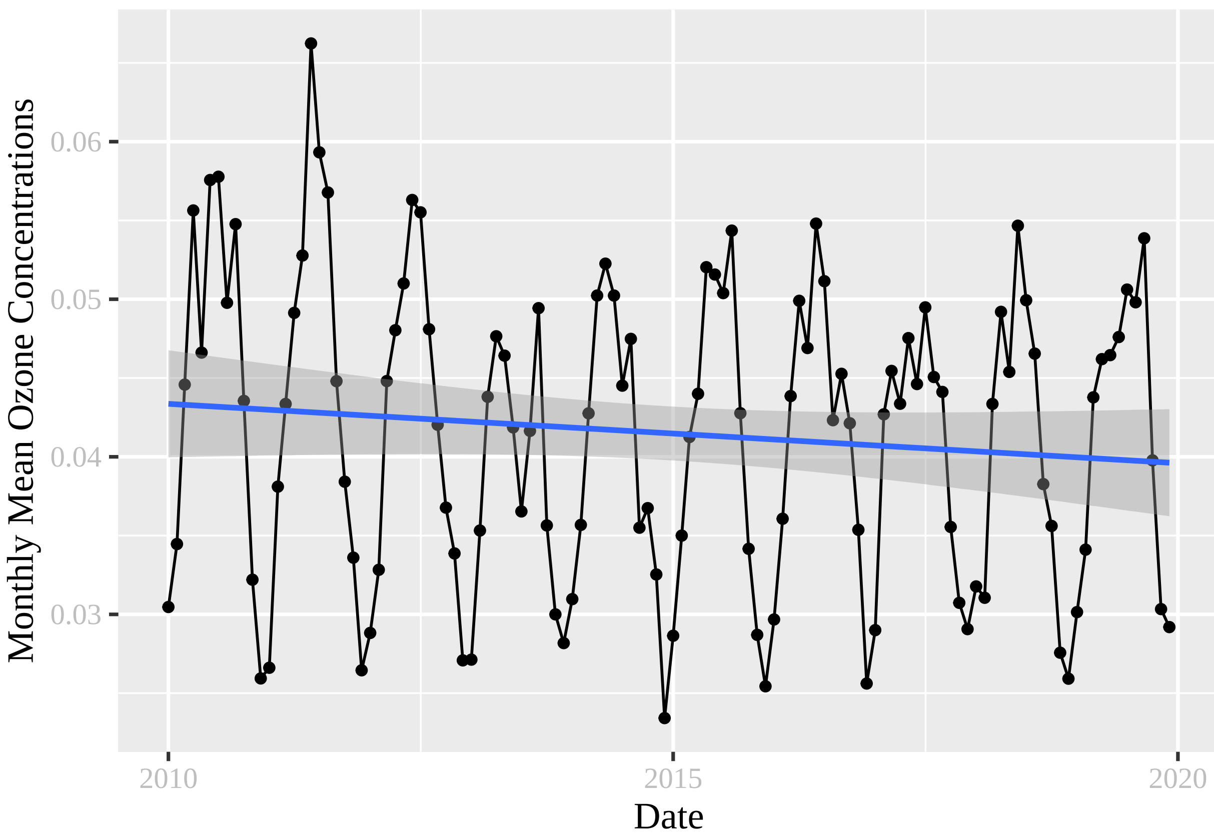
> Answer: Because our data displayed seasonal and/or cyclical trends when we plotted ozone concentrations over time.

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

mean_monthly_ozone_bytime <- ggplot(
  GaringerOzone_monthly_complete.df, aes(
    x = Date,
    y = mean.ozone.concentrations)) +
  geom_point() +
  geom_line() +
  geom_smooth(method = "lm") +
  labs(
    x = "Date",
    y = "Monthly Mean Ozone Concentrations") +
```

```
  mytheme1

print(mean_monthly_ozone_bytime)
```

```
## `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: At this station, Ozone concentrations do display a monotonic trend (p=0.046724), suggesting the trend does change, perhaps decrease (suggested by the negative tau = -0.143), throughout time, which is also suggested by the graph. The data does exhibit an seasonal and/or cyclical nature, suggesting the trend in ozone concentrations did change at this station.

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.ts <- as.data.frame(
  Ozone_monthly_decomp$time.series[,1:3])
```

```
GaringerOzone_monthly_Components.ts <- mutate(
  GaringerOzone_monthly_Components.ts, Observed = GaringerOzone_monthly_complete.df$mean.ozone.concentra
        Date = GaringerOzone_monthly_complete.df$Date)


#16

Kendall::MannKendall(GaringerOzone_monthly_Components.ts$Observed)
```

```
## tau = -0.0594, 2-sided pvalue =0.33732
```

Answer: The test on the non-seasonal Ozone monthly series suggests that there is not a monotonic trend, suggesting the ozone concentrations does not increase or decrease across time (p=0.33732). Compared to the previous test which suggested that the ozone concentrations did change over time.