Assignment 8: Time Series Analysis

Grace Randall

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

```
getwd()
```

```
## [1] "/home/guest/module1/EDE_Fall2023"
```

```
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
```

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

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
```

```
##     as.Date, as.Date.numeric
```

```r
library(trend)

my_theme <-
  theme(
    #line =              element_line(),
    #rect =              element_rect(),
    #text =              element_text(),

    # Modified inheritance structure of text element
    plot.title =        element_text(color = "midnightblue",hjust = 0.5),
    axis.title.x =      element_text(color = "midnightblue"),
    axis.title.y =      element_text(color = "midnightblue",angle = 90,
                                     vjust = 0.5, hjust=1),
    #axis.text =         element_text(),

    # Modified inheritance structure of line element
    #axis.ticks =        element_line(),
    panel.grid.major =  element_line(color="white"),
    #panel.grid.minor =  element_blank(),

    # Modified inheritance structure of rect element
    #plot.background =   element_rect(),
    panel.background =  element_rect(fill = "lightskyblue1"),
    #legend.key =        element_rect(),

    # Modifiying legend.position
    #legend.position = 'top',

    #complete = TRUE
  )
theme_set(my_theme)
```

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
file_list <-
  list.files("./Data/Raw/Ozone_TimeSeries", pattern=".csv", full.names=T)
GaringerOzone_raw <- do.call("rbind", lapply(file_list, read.csv))

dim(GaringerOzone_raw)
```

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

```r
head(GaringerOzone_raw)
```

```
##         Date Source   Site.ID POC Daily.Max.8.hour.Ozone.Concentration UNITS
## 1 01/01/2010    AQS 371190041   1                                0.031   ppm
## 2 01/02/2010    AQS 371190041   1                                0.033   ppm
## 3 01/03/2010    AQS 371190041   1                                0.035   ppm
## 4 01/04/2010    AQS 371190041   1                                0.031   ppm
## 5 01/05/2010    AQS 371190041   1                                0.027   ppm
```

```
## 6 01/07/2010    AQS 371190041   1                                0.033   ppm
##   DAILY_AQI_VALUE           Site.Name DAILY_OBS_COUNT PERCENT_COMPLETE
## 1              29 Garinger High School              17              100
## 2              31 Garinger High School              17              100
## 3              32 Garinger High School              17              100
## 4              29 Garinger High School              17              100
## 5              25 Garinger High School              17              100
## 6              31 Garinger High School              17              100
##   AQS_PARAMETER_CODE AQS_PARAMETER_DESC CBSA_CODE
## 1              44201              Ozone     16740
## 2              44201              Ozone     16740
## 3              44201              Ozone     16740
## 4              44201              Ozone     16740
## 5              44201              Ozone     16740
## 6              44201              Ozone     16740
##                           CBSA_NAME STATE_CODE          STATE COUNTY_CODE
## 1 Charlotte-Concord-Gastonia, NC-SC         37 North Carolina         119
## 2 Charlotte-Concord-Gastonia, NC-SC         37 North Carolina         119
## 3 Charlotte-Concord-Gastonia, NC-SC         37 North Carolina         119
## 4 Charlotte-Concord-Gastonia, NC-SC         37 North Carolina         119
## 5 Charlotte-Concord-Gastonia, NC-SC         37 North Carolina         119
## 6 Charlotte-Concord-Gastonia, NC-SC         37 North Carolina         119
##        COUNTY SITE_LATITUDE SITE_LONGITUDE
## 1 Mecklenburg       35.2401      -80.78568
## 2 Mecklenburg       35.2401      -80.78568
## 3 Mecklenburg       35.2401      -80.78568
## 4 Mecklenburg       35.2401      -80.78568
## 5 Mecklenburg       35.2401      -80.78568
## 6 Mecklenburg       35.2401      -80.78568
```

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 & 4
#done out of order so I could have a completely separate raw and processed
GaringerOzone_processed <-
  select(GaringerOzone_raw,
         "Date","Daily.Max.8.hour.Ozone.Concentration","DAILY_AQI_VALUE")
GaringerOzone_processed$Date <- mdy(GaringerOzone_processed$Date)

# 5
full_dates <- as.data.frame(
  seq(min(GaringerOzone_processed$Date),
```

```r
      max(GaringerOzone_processed$Date),1))
names(full_dates) <- "Date"

# 6
GaringerOzone <- left_join(full_dates,GaringerOzone_processed)
```

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

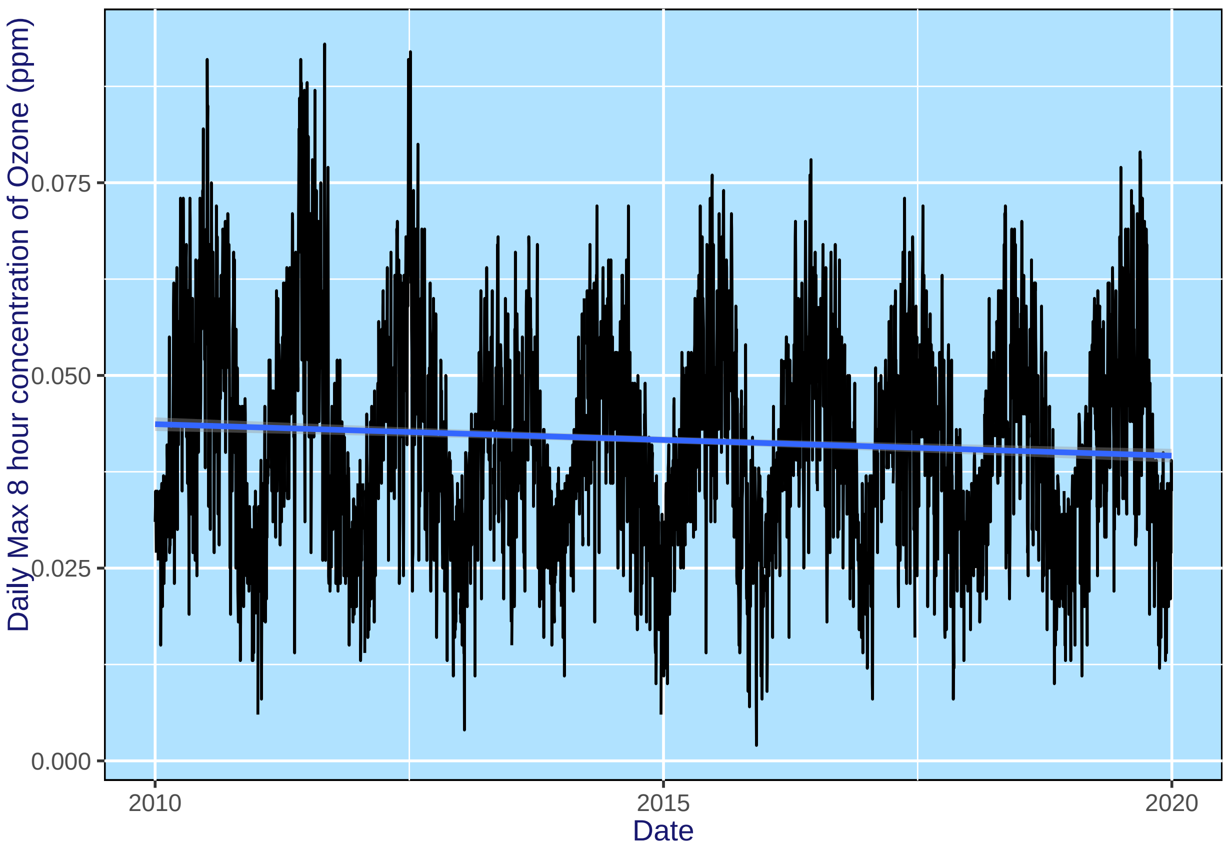
Visualize

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?

```r
#7
ggplot(GaringerOzone,aes(x=Date,y=Daily.Max.8.hour.Ozone.Concentration))+
  geom_line()+
  geom_smooth(method="lm")+
  ylab("Daily Max 8 hour concentration of Ozone (ppm) ")
```

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

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



Answer: there appears to be a slight downward trend in the concentration of ozone 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$Daily.Max.8.hour.Ozone.Concentration <-
  na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
GaringerOzone$DAILY_AQI_VALUE <-
  na.approx(GaringerOzone$DAILY_AQI_VALUE)
```

Answer: we use the liear model because it is the model that fits best with how concentrations of ozone are related to each other.

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(month=month(Date)) %>%
  mutate(year=year(Date)) %>%
  group_by(year, month) %>%
  summarise(meanOzone= mean(Daily.Max.8.hour.Ozone.Concentration),
            meanaqi = mean(DAILY_AQI_VALUE), ) %>%
  mutate(Date=dmy(paste("1",month,year)))
```

```
## `summarise()` has grouped output by 'year'. 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.

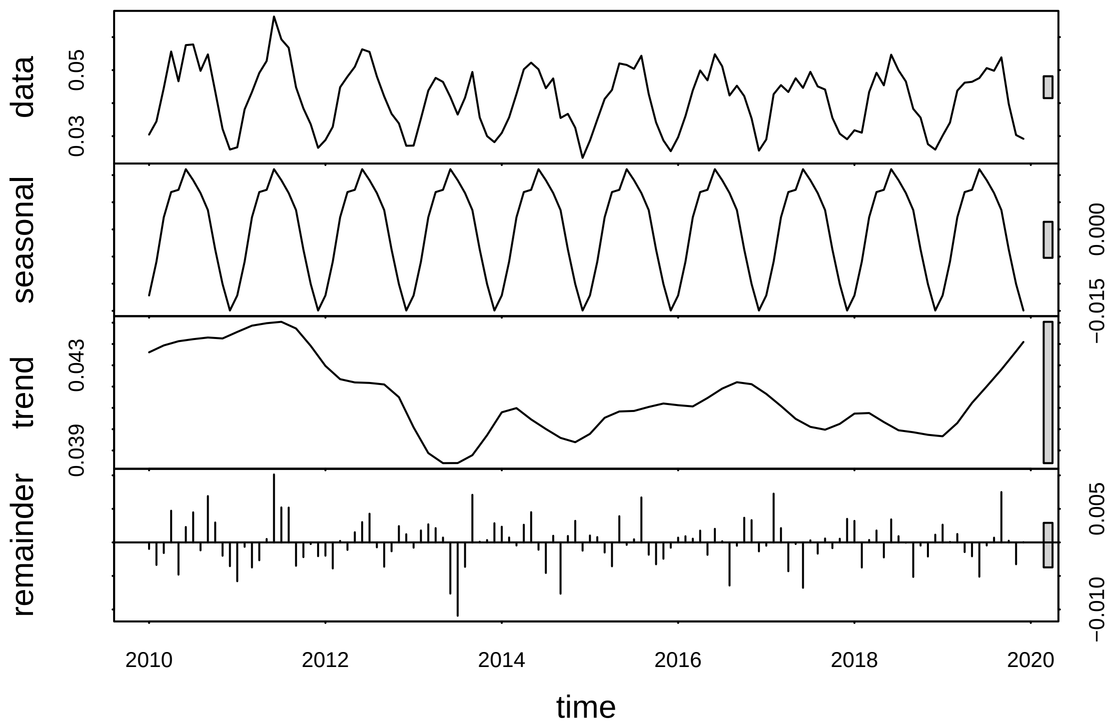
```
#10
GaringerOzone.daily.ts=ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                          start=min(GaringerOzone$Date),
                          end=max(GaringerOzone$Date),
                          frequency=1)

GaringerOzone.monthly.ts=ts(GaringerOzone.monthly$meanOzone,
                            start = c(2010,1),
                            end=c(2019,12),
                            frequency=12)
```

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

```
#11
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
monthly.trend <-  Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
monthly.trend2 <- smk.test(GaringerOzone.monthly.ts)

summary(monthly.trend)
```

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

```
summary(monthly.trend2)
```

```
##
##  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
```

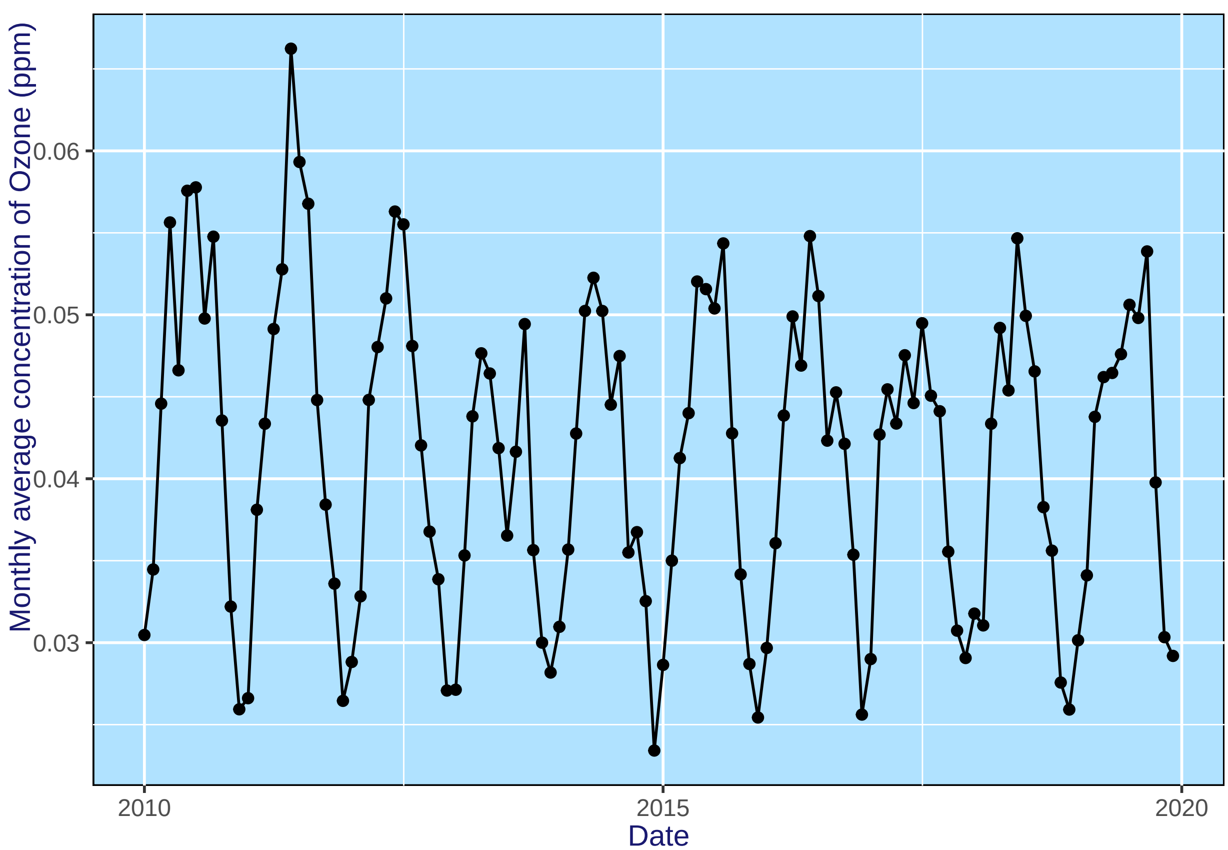
```
## Season 4:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 5:   S = 0  -15  125 -0.333 -1.252  0.21050
## Season 6:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 7:   S = 0  -11  125 -0.244 -0.894  0.37109
## Season 8:   S = 0   -7  125 -0.156 -0.537  0.59151
## Season 9:   S = 0   -5  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: This test is the most appropriate because it is the only one that accepts data that has a seasonal component.

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=Date,y=meanOzone))+
  geom_line()+
  geom_point()+
  ylab("Monthly average concentration of Ozone (ppm) ")
```



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 output of the test shows a p value of sighlty below 0.05. This means we can reject the null hypothesis and determine there is a trend. there are also some seasons that do not show much of a trend such as season 2 and 3. (Score = -77 , Var(Score) = 1499 denominator = 539.4972 tau = -0.143, 2-sided pvalue =0.046724)

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.monthlyComponents <-
  GaringerOzone.monthly.decomp$time.series[,1:3]

GaringerOzone.monthly.deseasoned <-
  GaringerOzone.monthlyComponents[,2] + GaringerOzone.monthlyComponents[,2]

#16

deseasoned.trend <-  Kendall::MannKendall(GaringerOzone.monthly.deseasoned)

summary(deseasoned.trend)
```

```
## Score =  -1922 , Var(Score) = 194366.7
## denominator =  7140
## tau = -0.269, 2-sided pvalue =1.3168e-05
```

Answer: When the seasonal commponent is removed, the p value becomes far lower making it easy to see there is a trend in the data.