Assignment 8: Time Series Analysis

Emma Kaufman

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

```r
getwd() #checking working directory
```

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

```r
#loading packages
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(agricolae)
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2023
```

```r
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
library(ggthemes)
library(lubridate)
library(trend)
library(zoo)
```

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

```r
# Set theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "forestgreen"),
        legend.position = "top",
        line = element_line(
          color = 'navyblue',
          linewidth = 0.5),
        plot.title = element_text(
          color = 'navyblue',
          size = 15),
        axis.title.x = element_text(
          color = "navyblue"),
        axis.title.y = element_text(
          color = "navyblue")
  )

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
#2
library(readr)
library(dplyr)

#set directory where files are located
csv_directory <- "~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/"
```

```
#empty list
data_frames <- list()

for (year in 2010:2019) {
  # file path for specific year
  file_path <- paste0(csv_directory, "EPAair_O3_GaringerNC", year, "_raw.csv")

  # read in file and append to list
  data <- read_csv(file_path)
  data_frames <- append(data_frames, list(data))
}

#single dataframe
GaringerOzone <- bind_rows(data_frames)
```

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
#using lubridate to change dates to date object
GaringerOzone$Date <- mdy(GaringerOzone$Date)

# 4
#selecting columns
GaringerOzone_subset <-GaringerOzone %>%
  rename(DailyMax8 = `Daily Max 8-hour Ozone Concentration`) %>%
  select(Date,DailyMax8,DAILY_AQI_VALUE)

# 5
#creating dataframe
Days <-as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "days"))
colnames(Days) <- "Date" #renaming the column

# 6
#left join to get all of the observations in Days
GaringerOzone <- Days %>%
  left_join(GaringerOzone_subset, by = "Date")
```

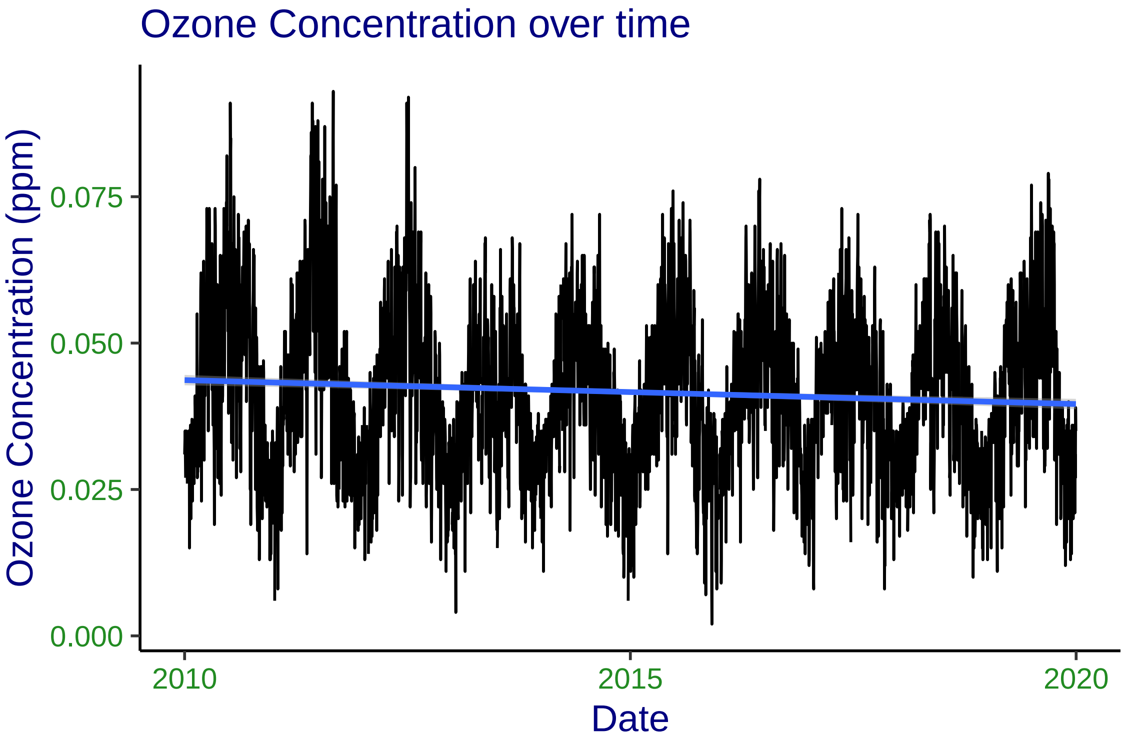
Visualize

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
PPM <- ggplot(GaringerOzone, aes(y = DailyMax8, x = Date)) +
  geom_line() +
  geom_smooth(method= "lm")+
 # axis.title.y = element_text(size = 6) +
  labs(y = "Ozone Concentration (ppm)") +
  labs(title = "Ozone Concentration over time")

plot(PPM)
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

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



Answer: It shows that ozone concentration has slightly decreased over time, it also shows yearly flucutations in ozone concentration. The slight decrease over time is evidence of a monotonic decreasing trend.

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
Ozone_clean <-
 GaringerOzone %>%
  mutate( DailyMax8.clean = zoo::na.approx(DailyMax8) )
```

> Answer: We didn't use a piecewise constant because if values are increasing or decreasing daily we want our estimate to represent that trend. If we just did nearest neighbors then our estimate wouldn't be as accurate, so we are using the linear interpolation because it takes the average of the known values before and after our missing value. Additionally there are only short periods of missing data, so linear interpolation is appropriate. We didn't use a spline interpolation because these data are best represent by a linear approximation rather than a quadratic one.

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
#manipulation for monthly dataframe
GaringerOzone.monthly <- Ozone_clean %>%
   mutate(month = month(Date)) %>%
   mutate(year = year(Date)) %>%
   group_by(year, month) %>%
   summarise(mean_ozone = mean(DailyMax8.clean))
```

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

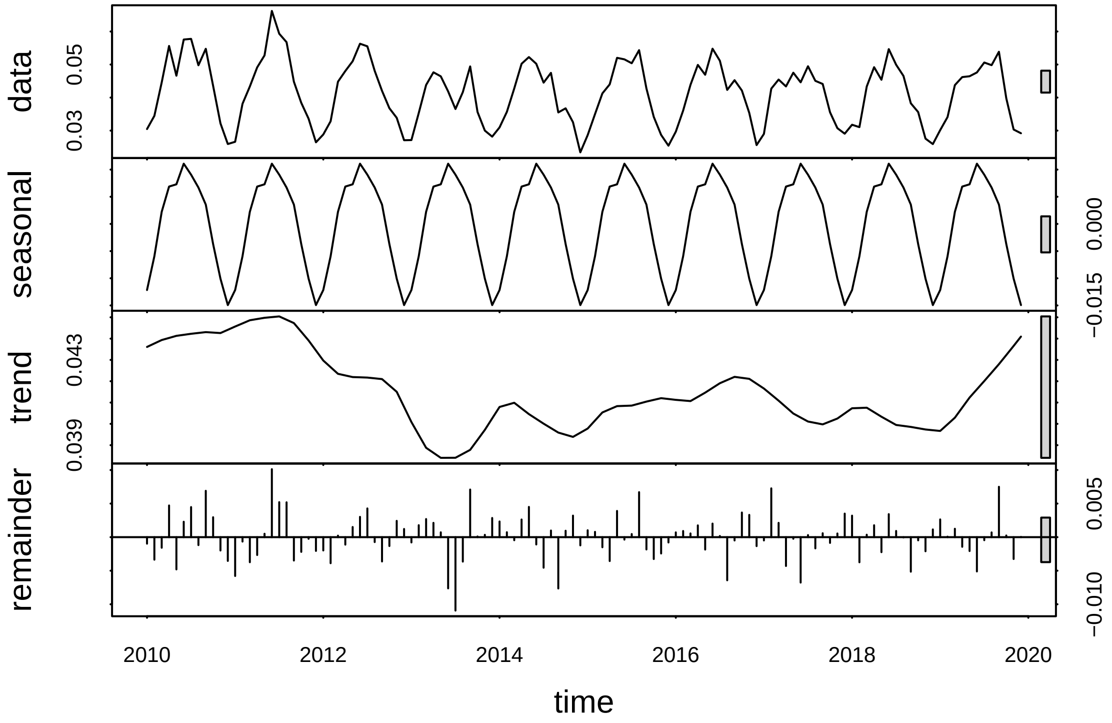
```
#making new Date column
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = make_date(year, month, 1))
```

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.

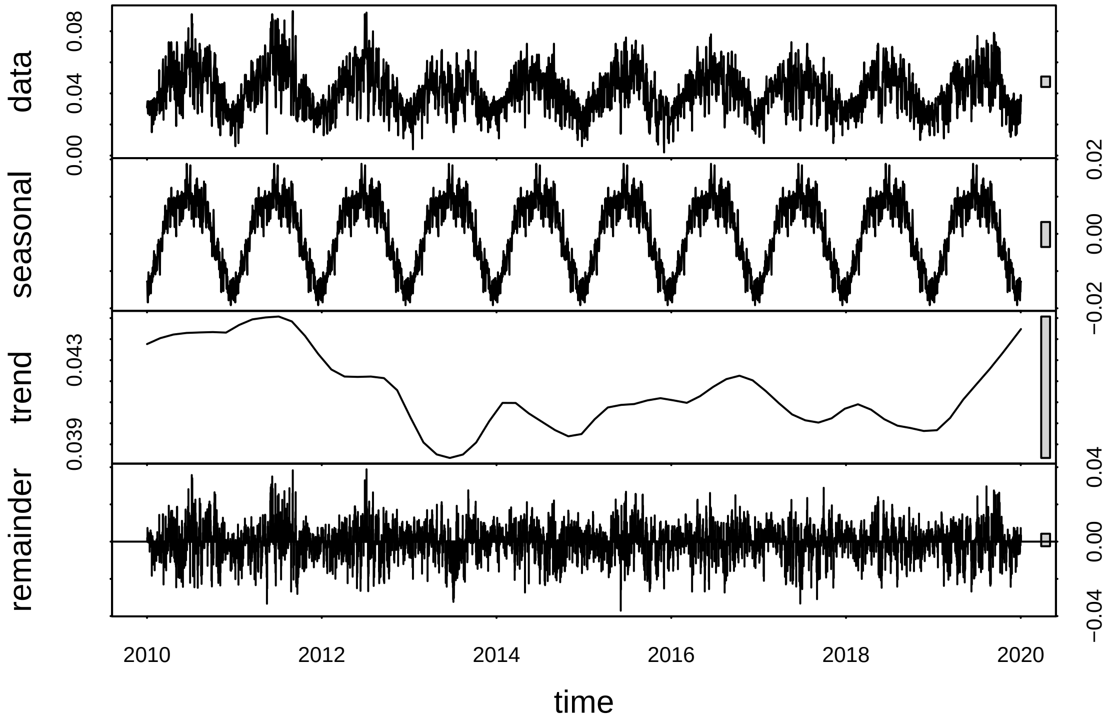
```
#10
#monthly ts
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone, start = c(2010,1), frequency = 12)

#daily ts
GaringerOzone.daily.ts <- ts(Ozone_clean$DailyMax8.clean, start=c(2010,1,1), frequency = 365)
```

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

```
#11
#decomposition and plotting for monthly and daily ts
Monthly_Decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(Monthly_Decomposed)
```



```
Daily_Decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(Daily_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 SMK test
Ozone_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Inspect results
Ozone_trend1
```

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

```
summary(Ozone_trend1)
```

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

```
Ozone_trend2 <- trend::smk.test(GaringerOzone.monthly.ts)
# Inspect results
Ozone_trend2
```

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

```
summary(Ozone_trend2)
```

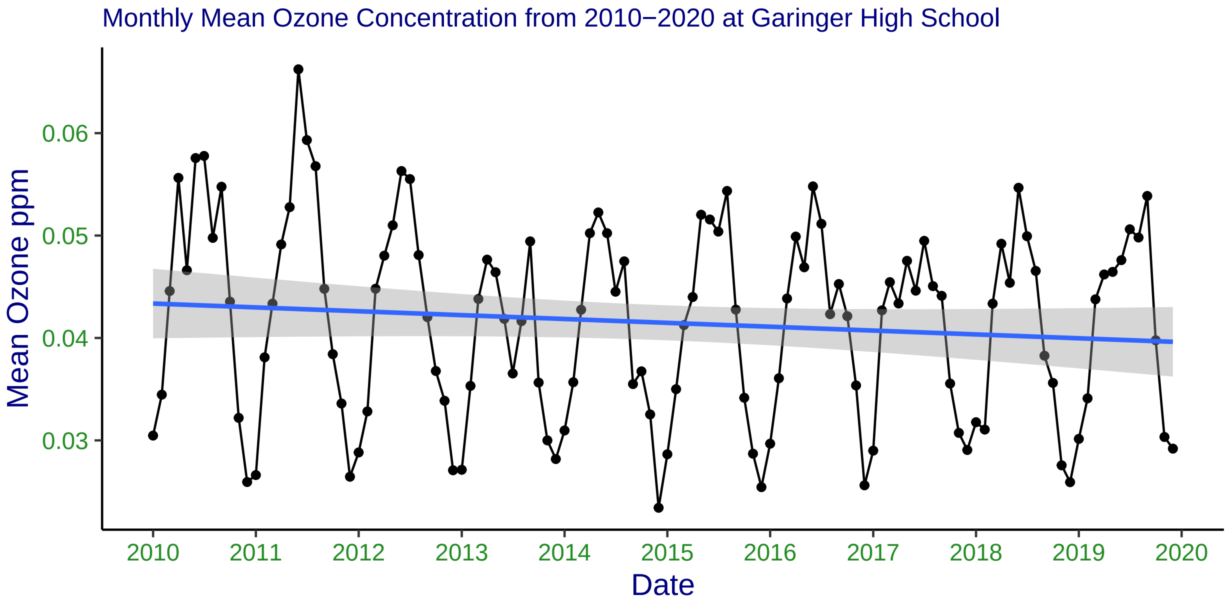
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                         S  varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## 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: The seasonal Mann Kendall trend analysis is appropriate because our data shows strong seasonality.This is the only test that is appropriate for seasonal data (unless we were to remove the seasonality from our data, in that case we could then try to run another test). Additionally while our data is not linear, we still sees a general monotonic trend (seen in the linear regression on the plot from question 7).

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
#Visualization
Ozone_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_ozone)) +
  geom_point() +
  geom_line() +
  ylab("Mean Ozone ppm") +
  geom_smooth( method = lm )+
  scale_x_date(date_breaks = "1 year", date_labels = '%Y')+
  labs(title = "Monthly Mean Ozone Concentration from 2010-2020 at Garinger High School")+
  theme(plot.title = element_text(size = 12))
print(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: The null hypothesis of the seasonal mann kendall is that the data are stationary (have not changed over the 2010s at this station). We get a p-value that is just below 0.05 (p= 0.04965), which means we can reject the null hypothesis and the trend we see in our data is statistically significant. So we can say that ozone concentrations have changed over the 2010s at this station (p=0.04965_. When looking at each indivdual season we see that some individual seasons don’t show an overall decreasing trend, but that the majority do have a large negative S value (indicating a decreasing trend). We see this negative trend on our plot with the negative slope for the linear regression line in blue, which indicates a decrease in monthly mean ozone concentration over time. The significance of this trend is confirmed by our p-value of 0.0497.

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

#decomposed time series into a dataframe
Garinger.late_Components <- as.data.frame(Monthly_Decomposed$time.series[,1:3])

#taking the dataframe and creating a column that removes seasonality
Garinger.late_Components_no_seasonality <- mutate(Garinger.late_Components,
        Mean_ozone = GaringerOzone.monthly$mean_ozone,
        Date = GaringerOzone.monthly$Date,
        no_seasonal_mean = Mean_ozone- seasonal)

#converting the dataframe back into a timeseries
no_seasonality.ts <- ts(Garinger.late_Components_no_seasonality$no_seasonal_mean,
                        start = c(2010,1),
                        frequency = 12)

#16

#running the MK test on the nonseasonal data
Ozone_nonseasonal <- Kendall::MannKendall(no_seasonality.ts)
summary(Ozone_nonseasonal)
```

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

> Answer: The p-value from the non-seasonal Mann Kendall test is 0.007. This means the trend in the data is strongly statistically significant (our p-value is much less than 0.05) for the non-seasonal data. Our p-value was 0.04965 for the seasonal data, so removing the seasonality resulted in a more significant monotonic trend in the data. The tau value of -.165 means it is a decreasing monotonic trend (we also saw a decreasing monotonic trend with the SMK test). This means that mean ozone is decreasing over the 2010s.