Assignment 8: Time Series Analysis

Rachael Stephan

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

```
#1
#set up chunk options
knitr::opts_chunk$set(message = FALSE, warning = FALSE)

#load library
library(tidyverse); library(zoo); library(here);
library(lubridate); library(trend)

#check working directory
getwd()
```

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

```
here()
```

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

```r
#set theme
mytheme <- theme_bw(base_size = 10)+
  theme(axis.title = element_text(size = 10, hjust = 0.5),
        plot.title.position = "panel",
        legend.box = "vertical",
        legend.location = "plot",
        axis.gridlines = element_line(color = "grey", linewidth = 0.25),
        axis.ticks = element_line(color = "black", linewidth = 0.5))
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
#create a list of files to download and then merge them
GaringerOzone <- list.files(path="Data/Raw/Ozone_TimeSeries",
                            pattern="*.csv",
                            full.names=TRUE)%>%
  lapply(read_csv) %>%
  bind_rows
```

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
# change date format as appropriate and select wanted columns
GaringerOzone <- GaringerOzone %>%
  mutate(Date = mdy(Date)) %>%
  select(Date, `Daily Max 8-hour Ozone Concentration`, DAILY_AQI_VALUE)

# 5
# create data frame and rename the column
Days <- as.data.frame(seq(from = as.Date("2010-01-01"),
                          to = as.Date("2019-12-31"),
                          by = "day"))

colnames(Days) <- c("Date")
```

```r
# 6
# combine data frames and check dimensions
GaringerOzone <- left_join(Days, GaringerOzone, by = "Date")
dim(GaringerOzone)
```

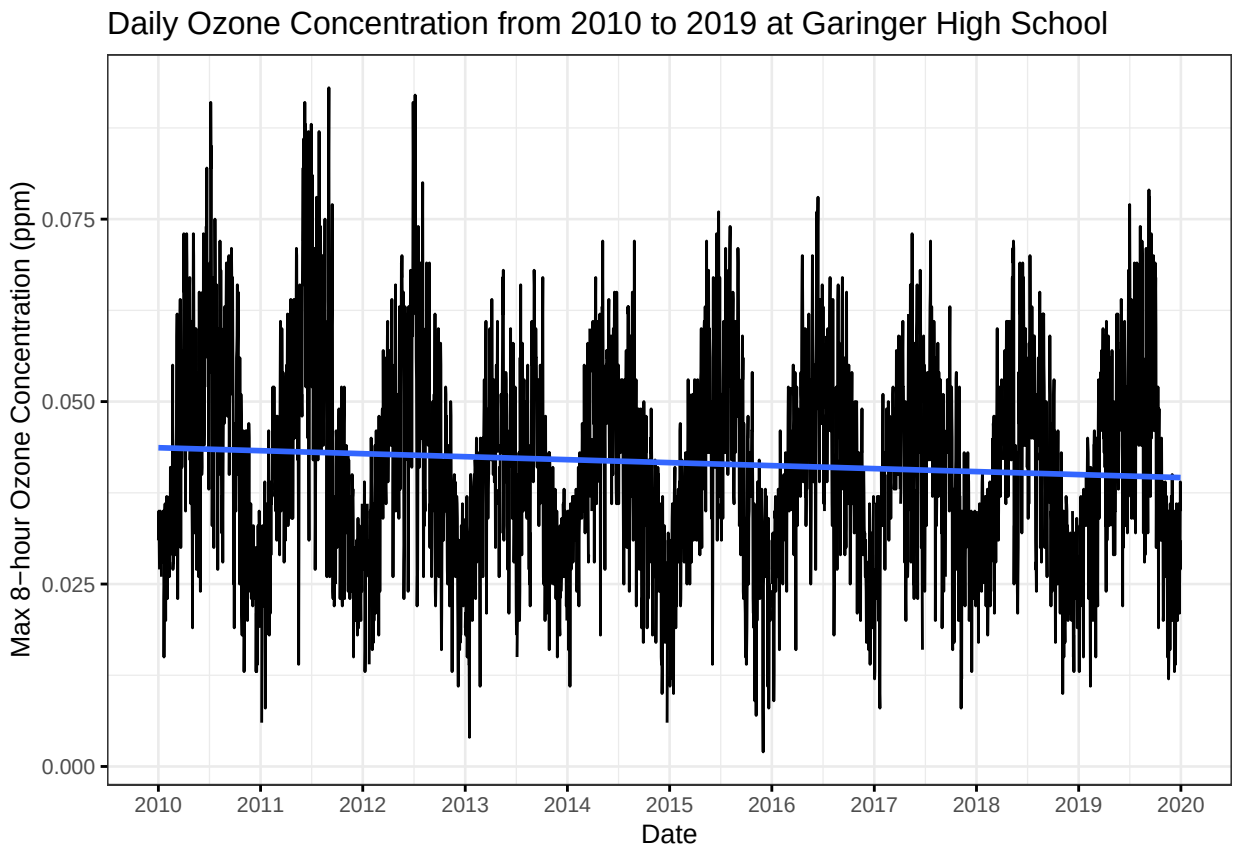
```
## [1] 3652    3
```

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?

```r
#7
lineplot <- ggplot(data = GaringerOzone,
                   aes(y = `Daily Max 8-hour Ozone Concentration`,
                       x = Date))+
  geom_line()+
  geom_smooth(method = "lm", se = F)+
  labs(title = "Daily Ozone Concentration from 2010 to 2019 at Garinger High School",
       y = "Max 8-hour Ozone Concentration (ppm)")+
  scale_x_date(breaks = seq(from = min(GaringerOzone$Date),
                            to = as.Date("2020-01-01"),
                            by = "1 year"),
               date_labels = "%Y")

lineplot
```



Answer: If there is a linear trend, it is a slight decrease in ozone concentrations over time. However, this trend is a small yearly decrease, and we have not tested for its significance.

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
# replace NAs
GaringerOzone$`Daily Max 8-hour Ozone Concentration` <-
  zoo::na.approx(GaringerOzone$`Daily Max 8-hour Ozone Concentration`)
```

Answer: A piecewise interpolation takes the nearest neighbour. In a timeseries, both neighbours are equally near. This causes an issue in which neighbour to refer to. There can also be large variation in mean ozone over small time periods (days). Piecewise variation will not allow for the timeseries to interpolate a middle value, and may falsely interpolate a continuing high or low daily ozone concentration. A spline interpolation creates a quadratic curve to interpolate between points. However, trends and seasonality occur over longer periods than the gaps in the timeseries. At small scale interpolation, there are not trends/curves to be interpolated due to the differences in values from day to day. Therefore, a linear connection bridging the gaps makes the most sense.

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
# use yearmon to turn into monthly data then back to date. Use date to group and
# summarize. Then, add year and month columns
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(Date = as.Date(as.yearmon(Date))) %>%
  group_by(Date) %>%
  summarize(MeanOzone = mean(`Daily Max 8-hour Ozone Concentration`)) %>%
  mutate(Year = year(Date)) %>%
  mutate(Month = month(Date)) %>%
  select(Date, Year, Month, MeanOzone)
```

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.

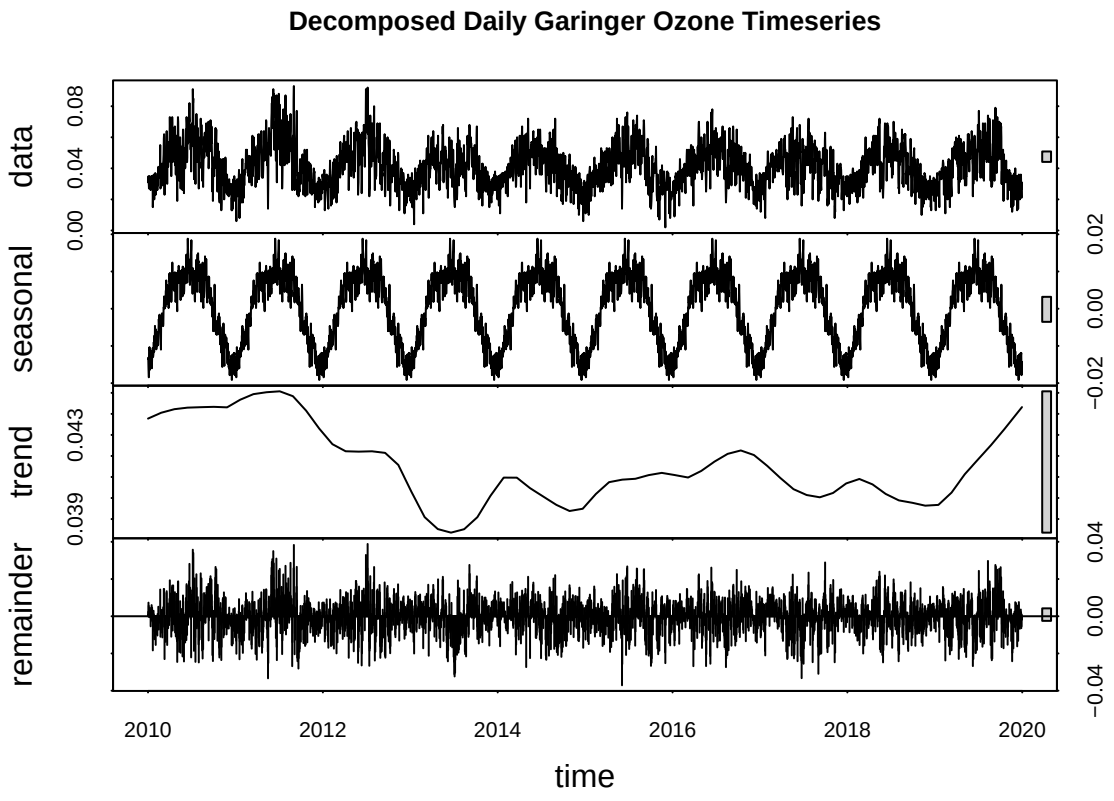
```
#10
# make a timeseries and retrieve the start and end times of the series for daily
# and monthly data
GaringerOzone.daily.ts <- ts(GaringerOzone$`Daily Max 8-hour Ozone Concentration`,
                             start = c(year(first(GaringerOzone$Date)),
                                       yday(first(GaringerOzone$Date))),
                             end = c(year(last(GaringerOzone$Date)),
                                     yday(last(GaringerOzone$Date))),
                             frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$MeanOzone,
                               start = c(first(GaringerOzone.monthly$Year),
                                         first(GaringerOzone.monthly$Month)),
                               end = c(last(GaringerOzone.monthly$Year),
                                       last(GaringerOzone.monthly$Month)),
                               frequency = 12)
```

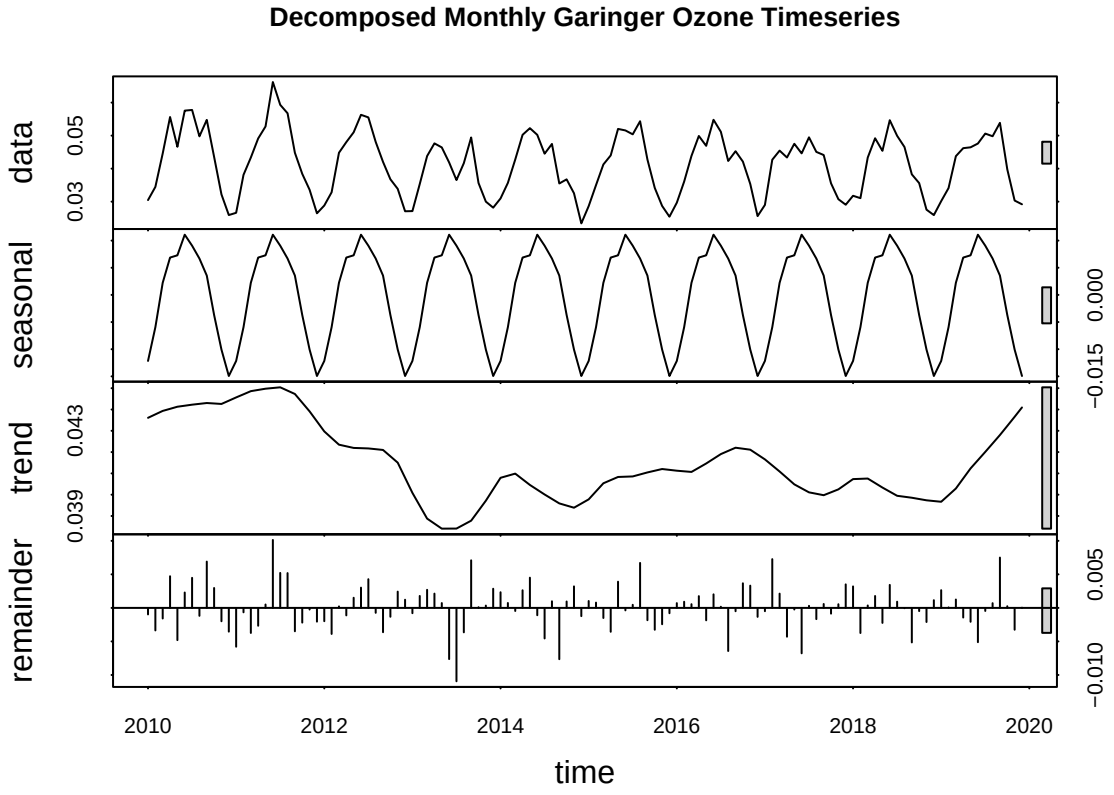
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11
#decompose the time series

GaringerOzone.daily.decomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily.decomp, main = "Decomposed Daily Garinger Ozone Timeseries")
```




```
GaringerOzone.monthly.decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(GaringerOzone.monthly.decomp, main = "Decomposed Monthly Garinger Ozone Timeseries")
```



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
GaringerOzone.monthly.smk <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

#view results
summary(GaringerOzone.monthly.smk)
```

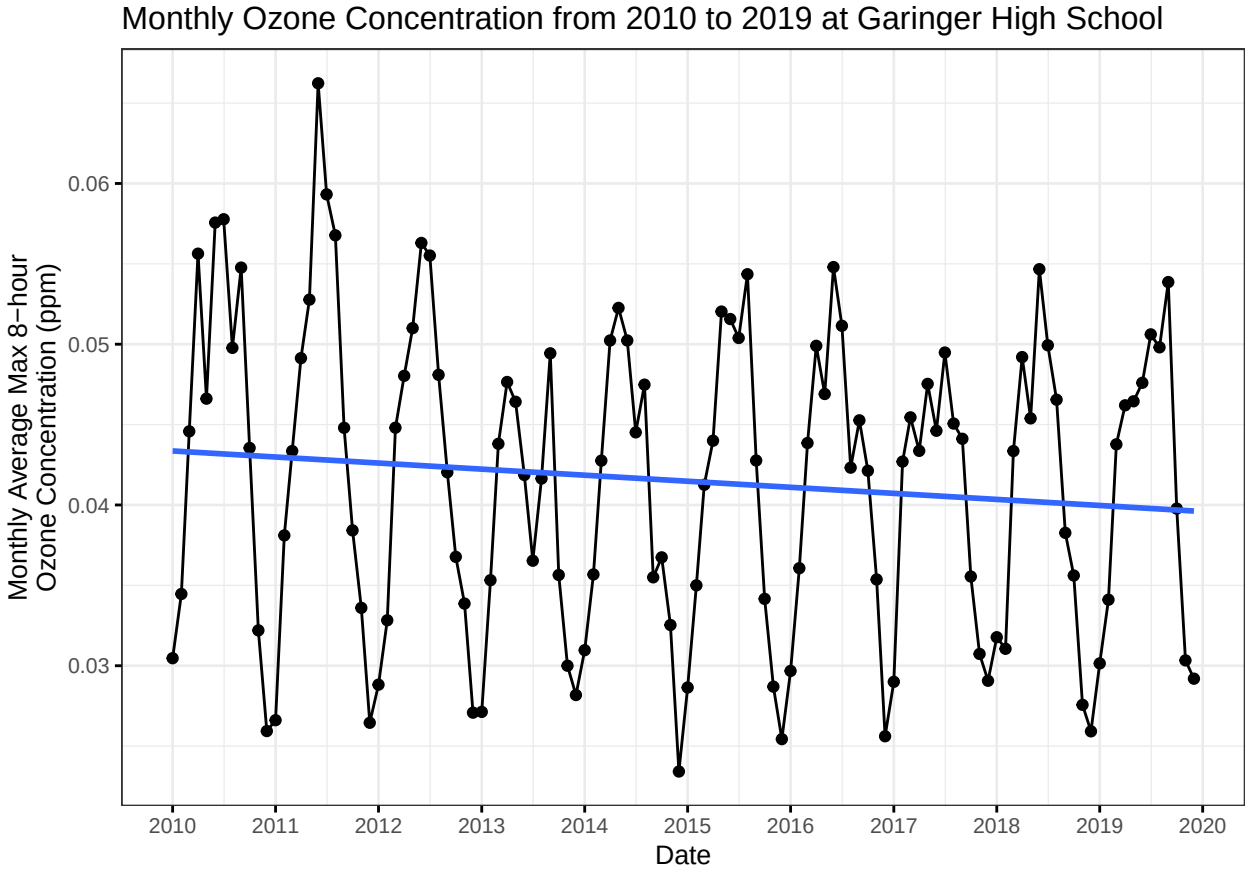
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The linear model was not used because we are not making assumptions about the data being parametric. The seasonal Mann-Kendall was used because it is the only test that could handle seasonality.

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.

```r
# 13
ozonetimeplot <- ggplot(GaringerOzone.monthly, aes(x=Date, y = MeanOzone))+
  geom_line()+
  geom_point()+
  labs(title = "Monthly Ozone Concentration from 2010 to 2019 at Garinger High School",
       y = "Monthly Average Max 8-hour\nOzone Concentration (ppm)")+
  scale_x_date(breaks = seq(from = first(GaringerOzone.monthly$Date),
                            to = as.Date("2020-01-01"),
                            by = "1 year"),
               date_labels = "%Y") +
  geom_smooth(method = lm, se = F)+
  theme(plot.margin = margin(l=4, unit = "pt"))

ozonetimeplot
```



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 study question was: Have ozone concentrations changed over the 2010s at this station? For changes in ozone concentrations, we will want to see a significant ($p<0.05$) monotonic trend. The graph shows clear seasonality through the monthly averages, and there is a slight decreasing linear trend. The monotonic trend is significant (p-value = 0.046724) and is a small negative trend (tau is negative and has a small magnitude, tau = -0.143). Therefore, we can conclude that there has been slight decreases to ozone concentrations at the station since 2010.

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
# Extract the components as a data frames
GaringerOzone.monthly.comp <- as.data.frame(GaringerOzone.monthly.decomp$time.series[,1:3])

#subtract seasonality from the total values
GaringerOzone.monthly.ts.deseason <- GaringerOzone.monthly.ts - GaringerOzone.monthly.comp$seasonal

#16
# run SMK test
GaringerOzone.monthly.mk <- Kendall::MannKendall(GaringerOzone.monthly.ts.deseason)

# view results
summary(GaringerOzone.monthly.mk)
```

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

Answer: Both tests show similar conclusions about ozone concentrations at Garinger High School from 2010 to 2019. Both tests have a p value less than 0.05 with a negative tau. This indicates that both analyses show there is a decreasing concentration in ozone at Garinger High School since 2010. The deseasoned test has a slightly larger tau magnitude and p-value. This indicates the analysis of the deseasoned data showed a slightly larger trend (although still small) with stronger evidence.