Assignment 8: Time Series Analysis

Gary Alvarez

Spring 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

```
## function (...)
## {
##     .root_env$root$f(...)
## }
## <bytecode: 0x55b928d2e908>
## <environment: namespace:here>
```

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
#Read all data sets
Gar2010 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
                    stringsAsFactors = TRUE)
Gar2011 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
                    stringsAsFactors = TRUE)
```

```
Gar2012 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
                    stringsAsFactors = TRUE)
Gar2013 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
                    stringsAsFactors = TRUE)
Gar2014 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
                    stringsAsFactors = TRUE)
Gar2015 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
                    stringsAsFactors = TRUE)
Gar2016 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
                    stringsAsFactors = TRUE)
Gar2017 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
                    stringsAsFactors = TRUE)
Gar2018 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
                    stringsAsFactors = TRUE)
Gar2019 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
                    stringsAsFactors = TRUE)


Garinger_Combined <- rbind(
  Gar2010, Gar2011, Gar2012, Gar2013, Gar2014,
  Gar2015, Gar2016, Gar2017, Gar2018, Gar2019)

# Check the dimensions of the combined dataframe
dim(Garinger_Combined)
```

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

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_subset <- Garinger_Combined %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

GaringerOzone_subset$Date <- mdy(GaringerOzone_subset$Date)

# 5
# Generate a sequence of dates from 2010-01-01 to 2019-12-31
dates <- seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "day")
```

```
# Convert the sequence of dates into a dataframe and rename the column
Days <- as.data.frame(dates)
names(Days) <- "Date"

# View the first few rows of the Days dataframe
head(Days)
```

```
##         Date
## 1 2010-01-01
## 2 2010-01-02
## 3 2010-01-03
## 4 2010-01-04
## 5 2010-01-05
## 6 2010-01-06
```

```
# 6
GaringerOzone <- left_join(Days, GaringerOzone_subset, by = "Date")

# Check the dimensions of the combined dataframe
dim(GaringerOzone)
```

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

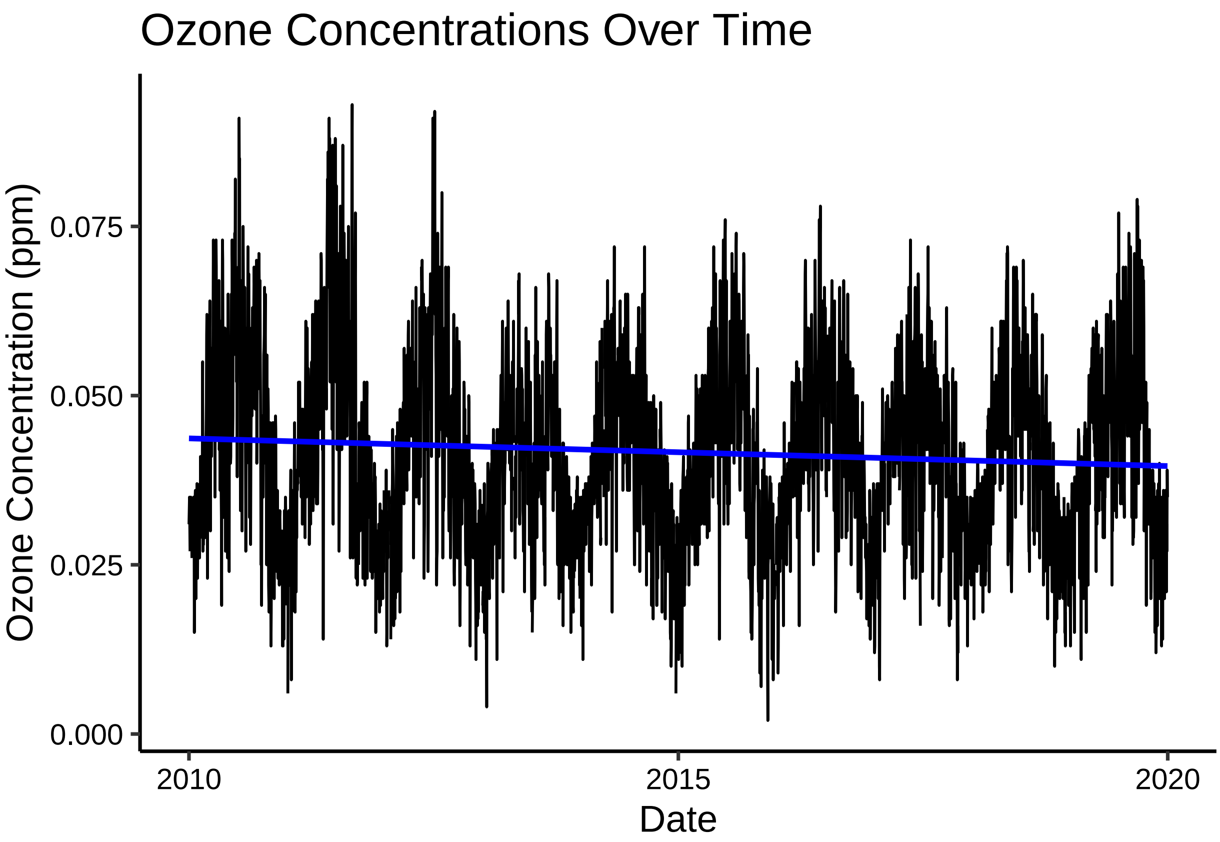
Visualize

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?

```
#7
plot_Ozone <- ggplot(
  GaringerOzone, aes(x = Date,
                     y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +  # Add a line plot
  geom_smooth(method = "lm",
              se = FALSE, color = "blue") +  # Add a smoothed line (linear trend)
  labs(x = "Date",
       y = "Ozone Concentration (ppm)",
       title = "Ozone Concentrations Over Time")
# Axis labels and title

plot(plot_Ozone)
```

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

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



Answer: Yes, the plot suggests a trend in Ozone concentration over time. Ozone concentration decreases over time, and we can also find seasonality in the plot (peaks and valleys repeat).

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,
  na.rm = FALSE)
```

Answer: Linear interpolation is used to find the estimated median, quartiles or percentiles of a dataframe. Both the piecewise constant and spline produce a smooth curve, which are not optimal to fill in the missing values in this case.

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
GaringerOzone.monthly <- GaringerOzone %>%
mutate(Year = year(Date)) %>%
mutate(Month = month(Date)) %>%
group_by(Month, Year) %>%
summarise(MeanOzone = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
mutate(Day="01") %>%
mutate(Date= as.Date(paste(Year, Month, Day, sep ="-"), "%Y-%m-%d")) %>%
dplyr::arrange(Date)
```

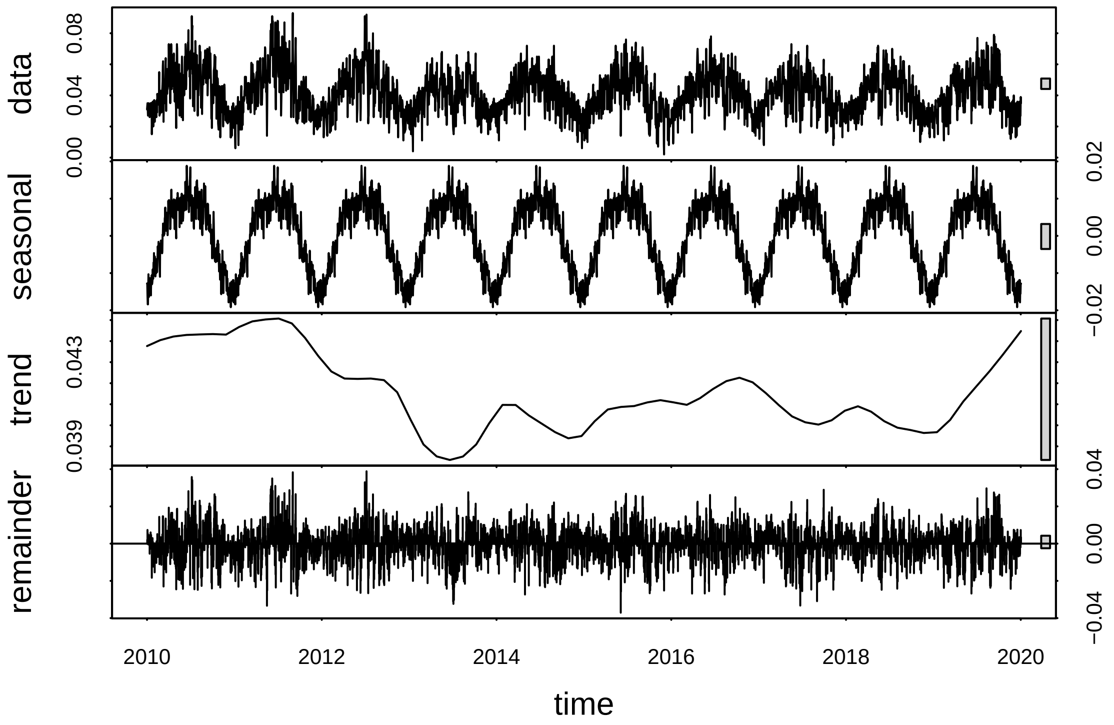
```
## `summarise()` has grouped output by 'Month'. You can override using the
## `.groups` argument.
```

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.

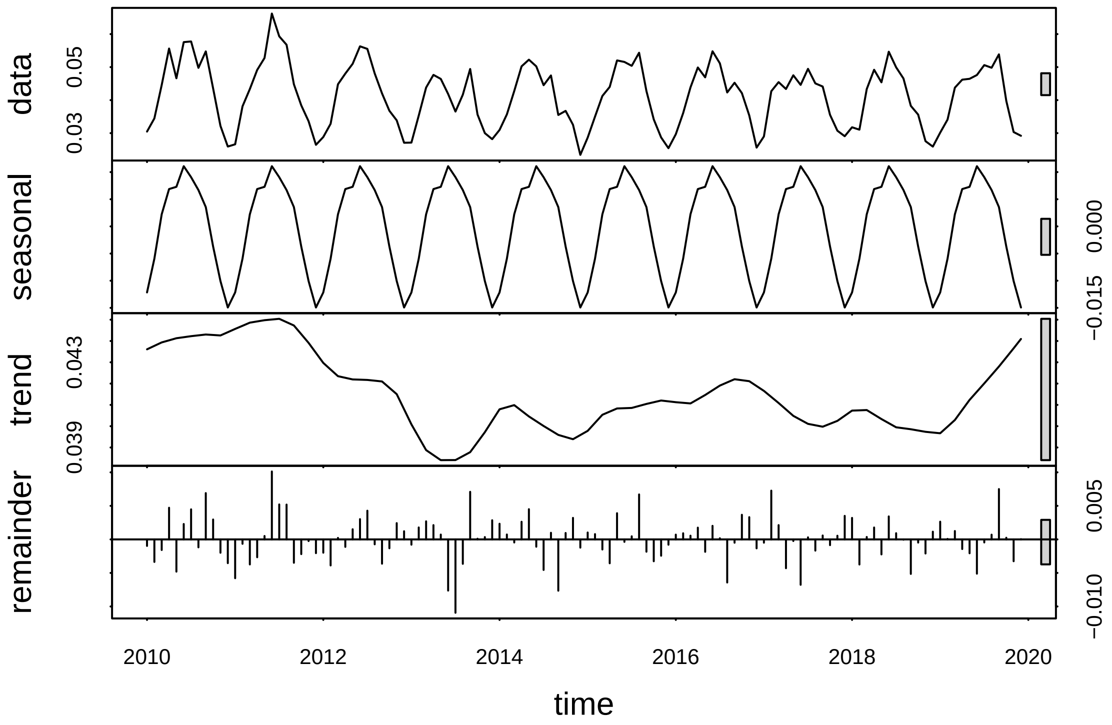
```r
#10
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
start=c(2010,1,1),
frequency=365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$MeanOzone,
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
GaringerOzone_dailydecomp <- stl(GaringerOzone.daily.ts,s.window = "periodic")
plot(GaringerOzone_dailydecomp)
```



```r
GaringerOzone_monthlydecomp <- stl(GaringerOzone.monthly.ts,s.window = "periodic")
plot(GaringerOzone_monthlydecomp)
```



12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

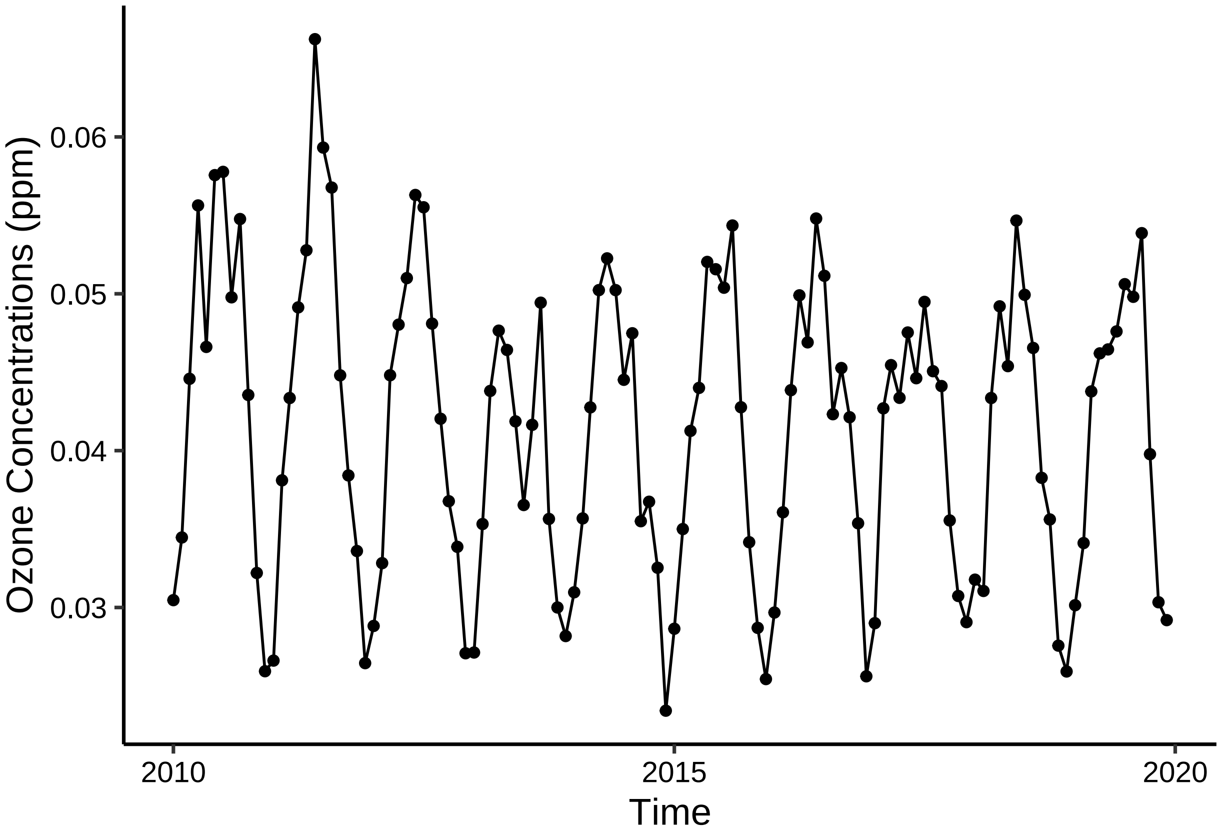
```
#12
Ozone_monthly_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

print((Ozone_monthly_trend))
```

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

Answer: The seasonal Mann-Kendall test is most appropriate in this case because we observed an evident wave pattern in the seasonal trend in the plots of Question #11.

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
MonthlyMeanOzone_bytime_lineplot <- GaringerOzone.monthly %>%
ggplot(aes(x= Date, y= MeanOzone)) +
geom_point() +
geom_line() +
labs(x= "Time", y="Ozone Concentrations (ppm)")
print(MonthlyMeanOzone_bytime_lineplot)
```



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.

```
Ozone_monthly_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
print((Ozone_monthly_trend))
```

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

Answer: Considering our research question, we have: H0: There is no change in ozone concentrations over the 2010s at this station. (The difference in the means of ozone=0) H1: There is change in ozone concentrations over the 2010s at this station. (The difference in the means of ozone!=0) In our seasonal Mann-Kendall test we obtained the values of tau = -0.143, and 2-sided pvalue =0.046724 (which is <0.05), so we can reject the null hypothesis in favor of the alternative, and conclude that there has been a change trend in ozone concentrations over the 2010s at this station, which matches the trend in our plot of Ozone Concentrations Over Time plot in #7.

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 <-
as.data.frame(GaringerOzone_monthlydecomp$time.series[,1:3])
GaringerOzone_monthly_deducted <-
GaringerOzone.monthly.ts - GaringerOzone_monthly_components$seasonal


#16
Ozone_monthly_nonseasonaltrend <-
Kendall::MannKendall(GaringerOzone_monthly_deducted)
print(Ozone_monthly_nonseasonaltrend)
```

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

> Answer: Compared to the results from the Seasonal Mann-Kendall test, after deduction the seasonal trend, the Mann Kendall test on the non-seasonal Ozone monthly series shows different values: tau = -0.165 (smaller than -0.143), and the 2-sided pvalue =0.0075402 (smaller than 0.046724). We can affirm that, excluding effects from the seasonal trend, there is a statistically significant decreasing trend in ozone concentrations over the 2010s at this station.