Assignment 7: Time Series Analysis

Analise Lindborg

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on time series analysis.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A07_TimeSeries.Rmd") prior to submission.

The completed exercise is due on Tuesday, March 16 at 11:59 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

```
## My document will not knit without this code
knitr::opts_knit$set(root.dir = '/Users/analiselindborg/Desktop/Desktop - Analise's MacBook Pro/Data Ana
```

```
#1
getwd()
```

```
## [1] "/Users/analiselindborg/Desktop/Desktop - Analise's MacBook Pro/Data Analytics/Environmental_Data
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.5     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'

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

```
library(trend)

theme <-
  theme_bw() +
  theme(panel.grid.minor = element_blank(),
        text = element_text(color = "black", size = 10),
        axis.text.x = element_text(color = "black"),
        axis.text.y = element_text(color = "black"))

theme_set(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.

```
#2
file_names <- dir("/Users/analiselindborg/Desktop/Desktop - Analise's MacBook Pro/Data Analytics/Enviro
GaringerOzone <- do.call(rbind,lapply(file_names,read.csv, stringsAsFactors = TRUE))
```

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$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

# 4
GaringerOzone.sub <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "1 days"))
```

```r
Days <- Days %>%
  rename(Date = `seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "1 days")`)

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

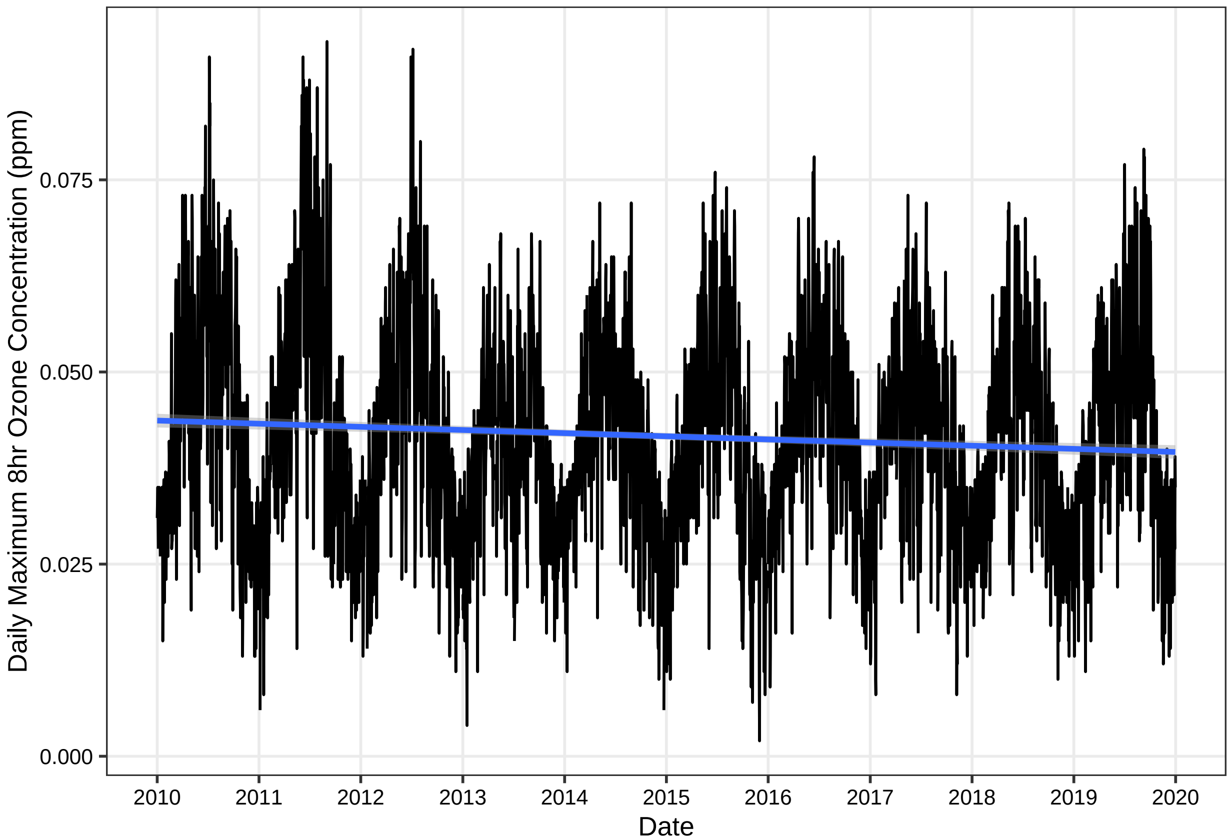
Visualize

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") +
  scale_x_date(date_breaks = "1 years", date_labels = "%Y") +
  labs(x = "Date", y = "Daily Maximum 8hr Ozone Concentration (ppm)") +
  theme
```

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

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



Answer:The plot suggests there may be a slight decreasing trend in concentraitons 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.interp <- GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration) )
```

> Answer: We used a liner interpolation because only short periods of time are missing (single days here and there). It is reasonable to assume that missing day concentrations will fall somewhere between the previous and next day, so linear interpolation is the best option.

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.interp %>%
    mutate(month = month(Date),
         year = year(Date)) %>%
  group_by(year, month) %>%
  summarise(mean.ozone.conc = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

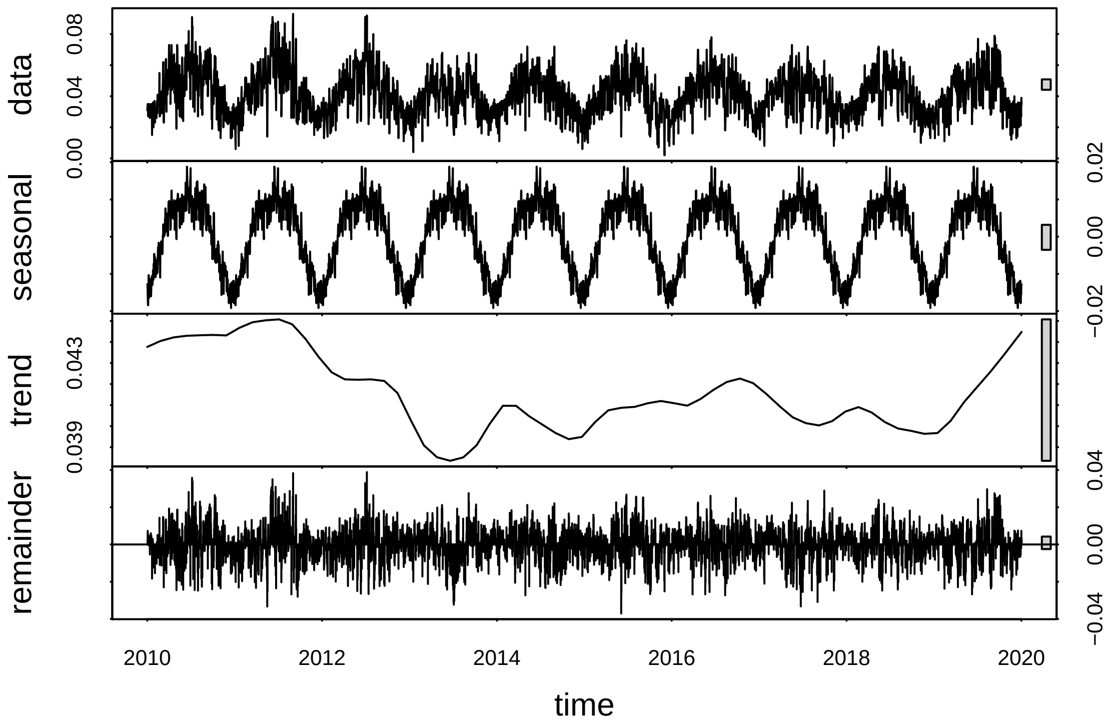
```
GaringerOzone.monthly$Date <- paste(GaringerOzone.monthly$month,
                                    "01",
                                    GaringerOzone.monthly$year,
                                    sep = "-" )

GaringerOzone.monthly$Date <- as.Date(GaringerOzone.monthly$Date, format = "%m-%d-%Y")
```

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.

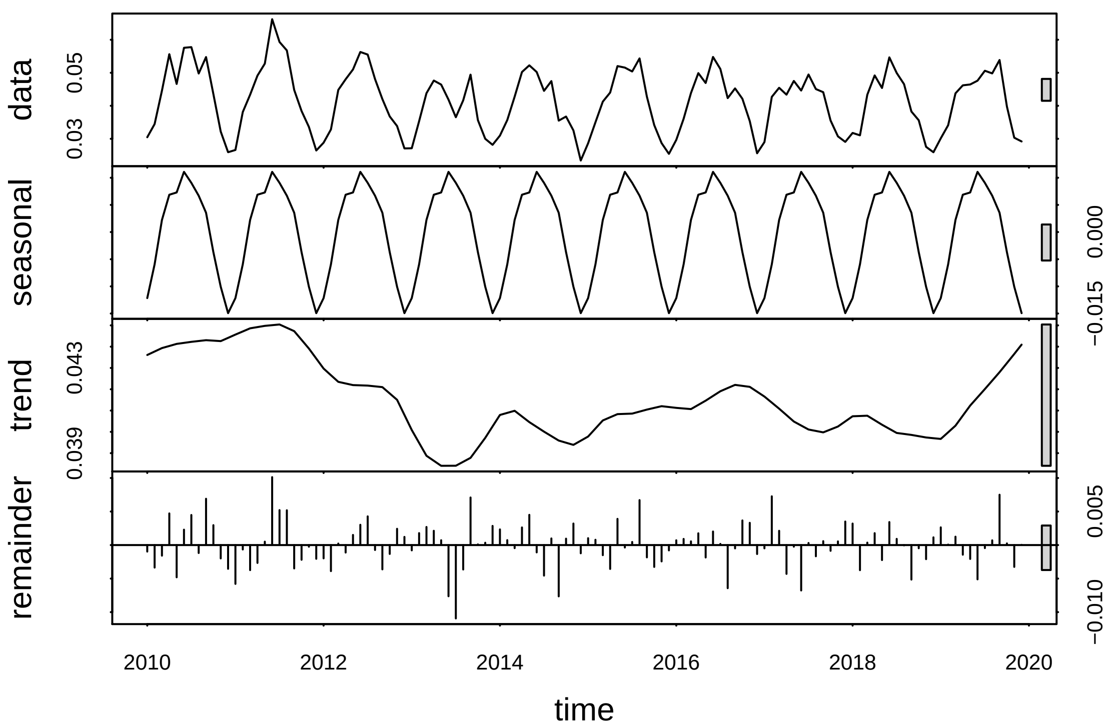
```
#10
GaringerOzone.daily.ts <- ts(GaringerOzone.interp$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010,1), frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean.ozone.conc,
                               start = c(2010,1), frequency = 12)
```

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

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

GaringerOzone.monthly.decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")

#plots
plot(GaringerOzone.daily.decomp)
```




```
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?

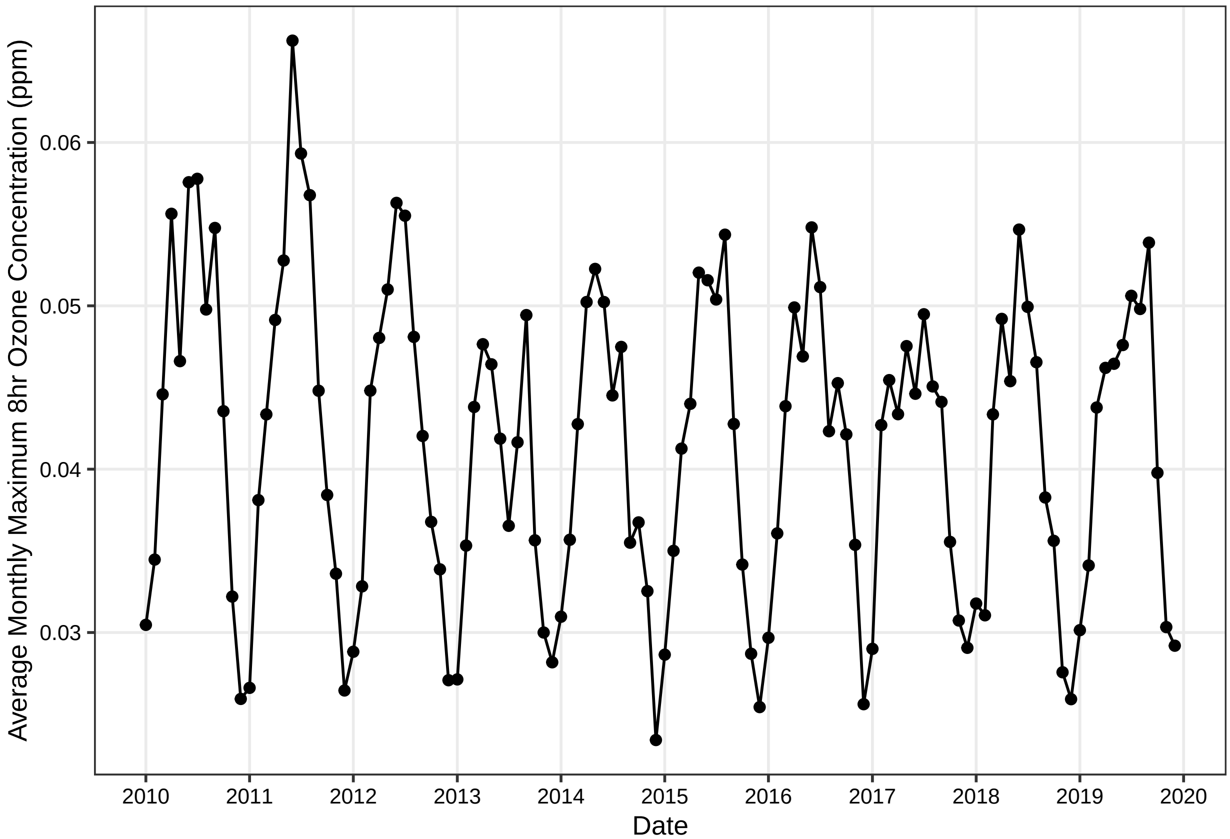
```
#12
trend.analysis <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(trend.analysis)
```

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

Answer: A seasonal Mann-Kendall is appropriate because we observe seasonality (concentration differences that fluctuate regularly with season).

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
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean.ozone.conc)) +
  geom_point()+
  geom_line()+
  scale_x_date(date_breaks = "1 years", date_labels = "%Y") +
  labs(x = "Date", y = "Average Monthly Maximum 8hr Ozone Concentration (ppm)") +
  theme
```



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: It appears that average monthly ozone concentrations at this station has changed significantly from 2010 to 2020. There is a slight decrease in ozone concentration over time (tau = -0.143), and it is significant based on a 95% significance level (p-value = 0.047).

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.monthly.noseason <- GaringerOzone.monthly.decomp$time.series[,2:3]

#16
no.season.trend <- Kendall::MannKendall(GaringerOzone.monthly.noseason)
summary(no.season.trend)
```

```
## Score =  -16300 , Var(Score) = 1545533
## denominator =  28680
## tau = -0.568, 2-sided pvalue =< 2.22e-16
```

> Answer: With seasonality removed, there is a much more signficant decrease in average monthly ozone concentration over time at this station (tau = -.568 and p-value <2.22e-16)