Assignment 7: Time Series Analysis

Elise Boos

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 Monday, March 14 at 7:00 pm.

Set up

1. Set up your session:

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

```
#1
#check wd
getwd()
```

```
## [1] "/Users/elise/Desktop/Data_Analytics/Environmental_Data_Analytics_2022/Assignments"
```

```
#load packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

```
## -- 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(trend)
library(zoo)
```

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

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

```
library(Kendall)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(dplyr)
library(plyr)
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```
library(readr)

#define and set theme
mytheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "bottom")
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.

```
#2
#load data in bulk and combine into single dataframe
GaringerOzone <- list.files(path = "../Data/Raw/Ozone_TimeSeries",
                      pattern = "*.csv", full.names = TRUE) %>%
 lapply(read.csv) %>%
 bind_rows

#check dimensions
dim(GaringerOzone)
```

```
## [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
#set date column as date class
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")
class(GaringerOzone$Date)
```

```
## [1] "Date"
```

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

```
##         Date Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
## 1 2010-01-01                                0.031              29
## 2 2010-01-02                                0.033              31
## 3 2010-01-03                                0.035              32
## 4 2010-01-04                                0.031              29
## 5 2010-01-05                                0.027              25
## 6 2010-01-07                                0.033              31
```

```
# 5
#create Days dataframe
Days <- data.frame(Date = seq.Date(from = as.Date("2010-01-01"),
                                   to = as.Date("2019-12-31"), by = "day"))
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
#combined dataframe
GaringerOzone <- left_join(Days, GaringerOzone_edit)
```

```
## Joining, by = "Date"
```

```
#dimensions
dim(GaringerOzone)
```

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

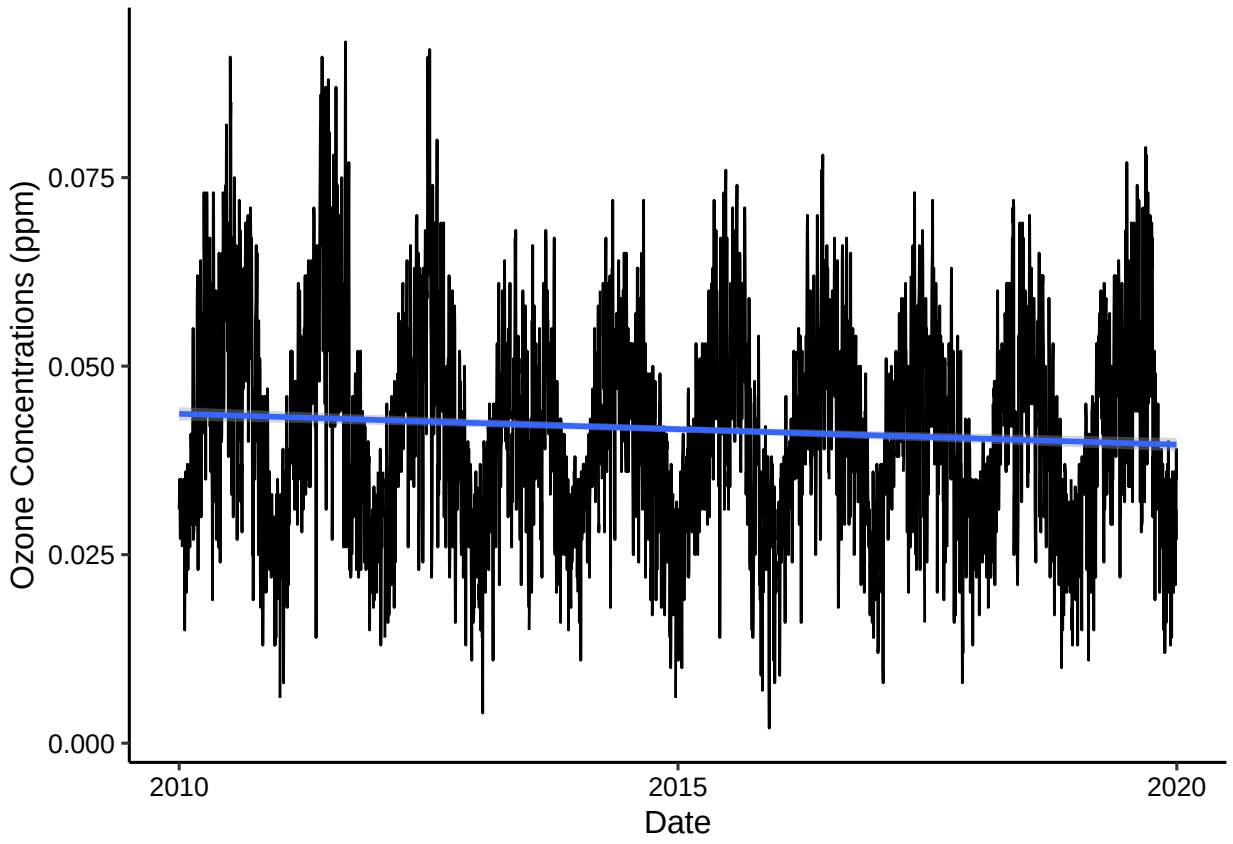
Visualize

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
ozone_plot <-
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  ylab("Ozone Concentrations (ppm)") +
  geom_smooth(method = lm)
print(ozone_plot)
```

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

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



Answer: There seems to be a slight decrease in ozone concentration 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
#create new column with linearly interpolated values
GaringerOzone_clean <-
  GaringerOzone %>%
  mutate(Ozone.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

head(GaringerOzone_clean$Ozone.clean)
```

```
## [1] 0.031 0.033 0.035 0.031 0.027 0.030
```

Answer: We didn't use piecewise constant because we don't want to assume the ozone concentration would be the same as the closest data value and we didn't use spline because we wanted to linearly connect the dots and there didn't appear to be a quadradic relationship between data values.

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
#create monthly mean concentration data set for each month
GaringerOzone.monthly <- GaringerOzone_clean %>%
  mutate(Month = month(Date), Year = year(Date)) %>%
  mutate(Month_Year = my(paste0(Month,"-",Year))) %>%
  dplyr::group_by(Month_Year) %>%
  dplyr::summarise(mean.monthly.ozone = mean(Ozone.clean))

head(GaringerOzone.monthly)
```

```
## # A tibble: 6 x 2
##   Month_Year mean.monthly.ozone
##   <date>                  <dbl>
## 1 2010-01-01             0.0305
## 2 2010-02-01             0.0345
## 3 2010-03-01             0.0446
## 4 2010-04-01             0.0556
## 5 2010-05-01             0.0466
## 6 2010-06-01             0.0576
```

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.

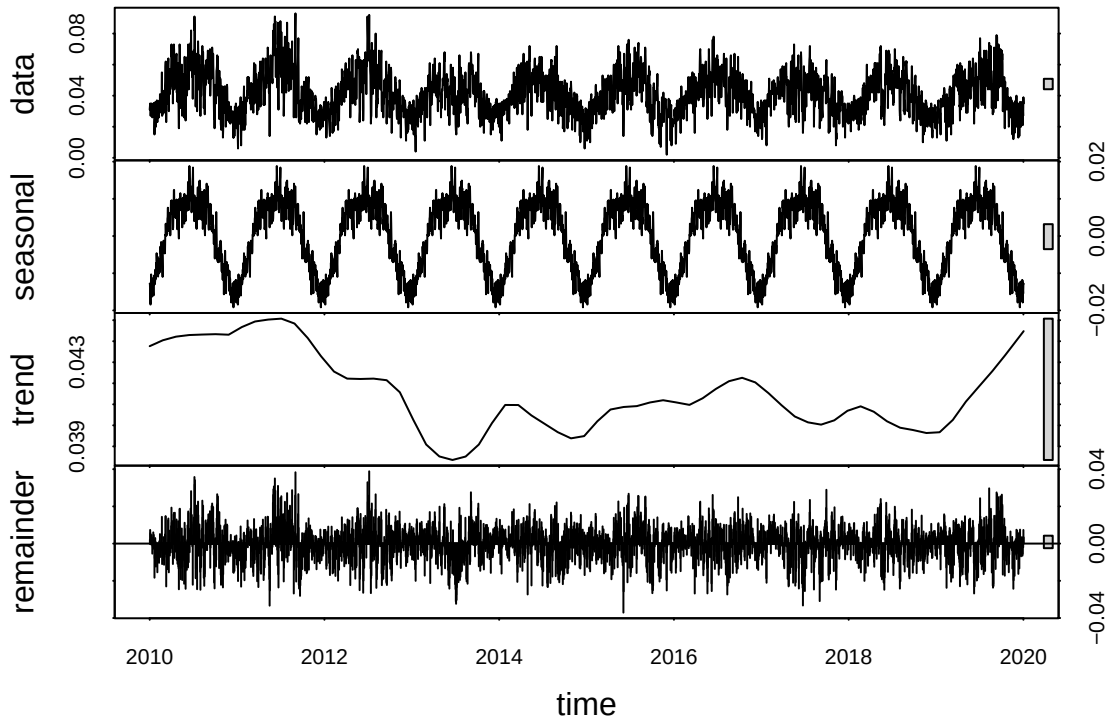
```
#10
#daily time series
GaringerOzone.daily.ts <- ts(GaringerOzone_clean$Ozone.clean,
                             start = c(2010,1), frequency = 365)

#monthly time series
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean.monthly.ozone,
                               start = c(2010,1), frequency = 12)
head(GaringerOzone.monthly.ts, 10)
```

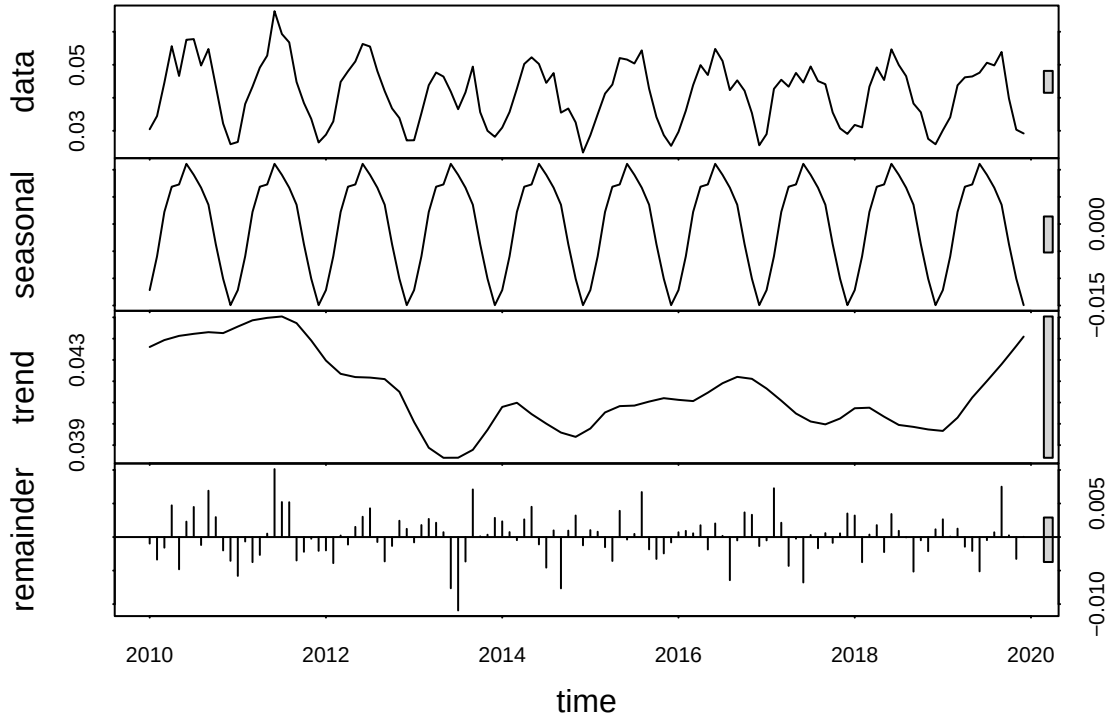
```
##  [1] 0.03046774 0.03446429 0.04458065 0.05563333 0.04661290 0.05756667
##  [7] 0.05777419 0.04977419 0.05476667 0.04354839
```

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

```
#11
#daily decomposed
GaringerOzone.daily.ts_Decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")
# Visualize the decomposed series.
plot(GaringerOzone.daily.ts_Decomposed)
```




```
#monthly decomposed
GaringerOzone.monthly.ts_Decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
# Visualize the decomposed series.
plot(GaringerOzone.monthly.ts_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?

```r
#12
#monotonic trend analysis
ozone_month_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
# Inspect results
ozone_month_trend
```

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

```r
summary(ozone_month_trend)
```

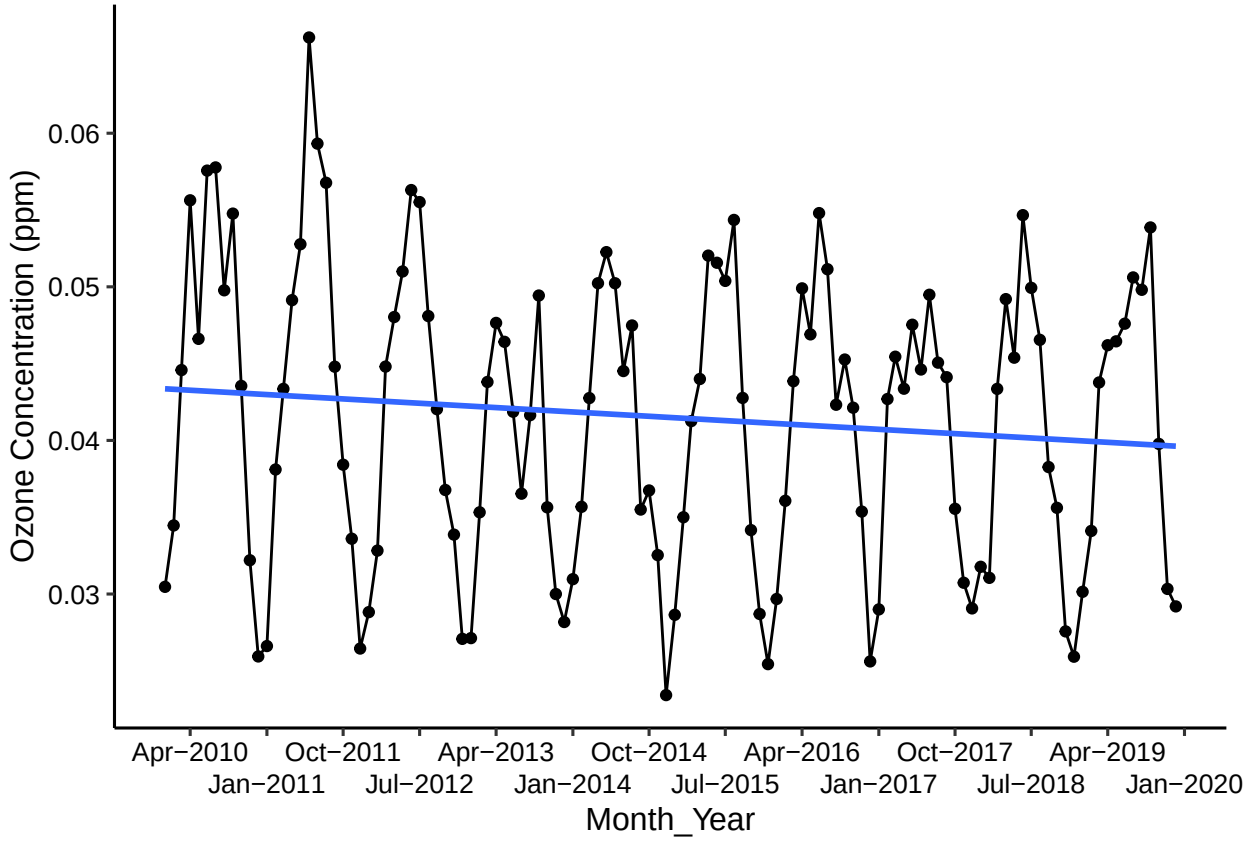
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: Because there is seasonal patterns/oscillations in the data.

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
month_plot <-
ggplot(GaringerOzone.monthly, aes(x = Month_Year, y = mean.monthly.ozone)) +
  geom_point() +
  geom_line() +
  scale_x_date(date_breaks = "9 months", date_labels = "%b-%Y") +
  guides(x = guide_axis(n.dodge = 2)) +
  geom_smooth(method = lm, se = FALSE) +
  ylab("Ozone Concentration (ppm)")
print(month_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: From the graph we can see that there are seasonal variations in ozone concentrations with higher concentrations in the summer months and lowest concentrations in winter months. Over time though, there is not a significant trend seen in ozone concentration over ($p = 0.163$).

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
#remove seasonal component
GaringerOzone.monthly_Components <- as.data.frame(GaringerOzone.monthly.ts_Decomposed$time.series[,2:3]
GaringerOzone.monthly_Components <- GaringerOzone.monthly_Components %>%
        mutate(Date = GaringerOzone.monthly$Month_Year) %>%
        mutate(data = trend + remainder) %>%
        select(Date, data)


#16
#run time series on new data with seasonal removed
f_month <- month(first(GaringerOzone.monthly_Components$Date))
```

```
f_year <- year(first(GaringerOzone.monthly_Components$Date))
month_nonseasonal_data_ts <- ts(GaringerOzone.monthly_Components$data,
                   start=c(f_year,f_month),
                   frequency=12)

month_trend <- MannKendall(month_nonseasonal_data_ts)
month_trend
```

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

```
summary(month_trend)
```

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

Answer: Now there is a significant trend of decline in ozone concentration (p = 0.0075) compared to when we ran the seasonal mannn kendall where we didn't get a significant p value.