Assignment 7: Time Series Analysis

Sam Vanasse

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
getwd()
```

```
## [1] "/Users/samvanasse/Desktop/ENV872-R/Environmental_Data_Analytics_2022/Assignments"
```

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

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

```
library(zoo)
```

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

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

```
library(trend)
library(lubridate)
```

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

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

```
library(dplyr)

def.theme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")

theme_set(def.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
Ozone.TimeSeries.2010 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", stri
Ozone.TimeSeries.2011 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", stri
Ozone.TimeSeries.2012 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", stri
Ozone.TimeSeries.2013 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", stri
Ozone.TimeSeries.2014 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", stri
Ozone.TimeSeries.2015 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", stri
Ozone.TimeSeries.2016 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", stri
Ozone.TimeSeries.2017 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", stri
Ozone.TimeSeries.2018 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", stri
Ozone.TimeSeries.2019 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", stri

GaringerOzone <- rbind(Ozone.TimeSeries.2010, Ozone.TimeSeries.2011, Ozone.TimeSeries.2012, Ozone.TimeS
```

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")
```

```r
# 4
GaringerOzone <-
  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 = "day")) %>%
  rename("Date" = `seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "day")`)

# 6
GaringerOzone <- left_join(Days, GaringerOzone)
```

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

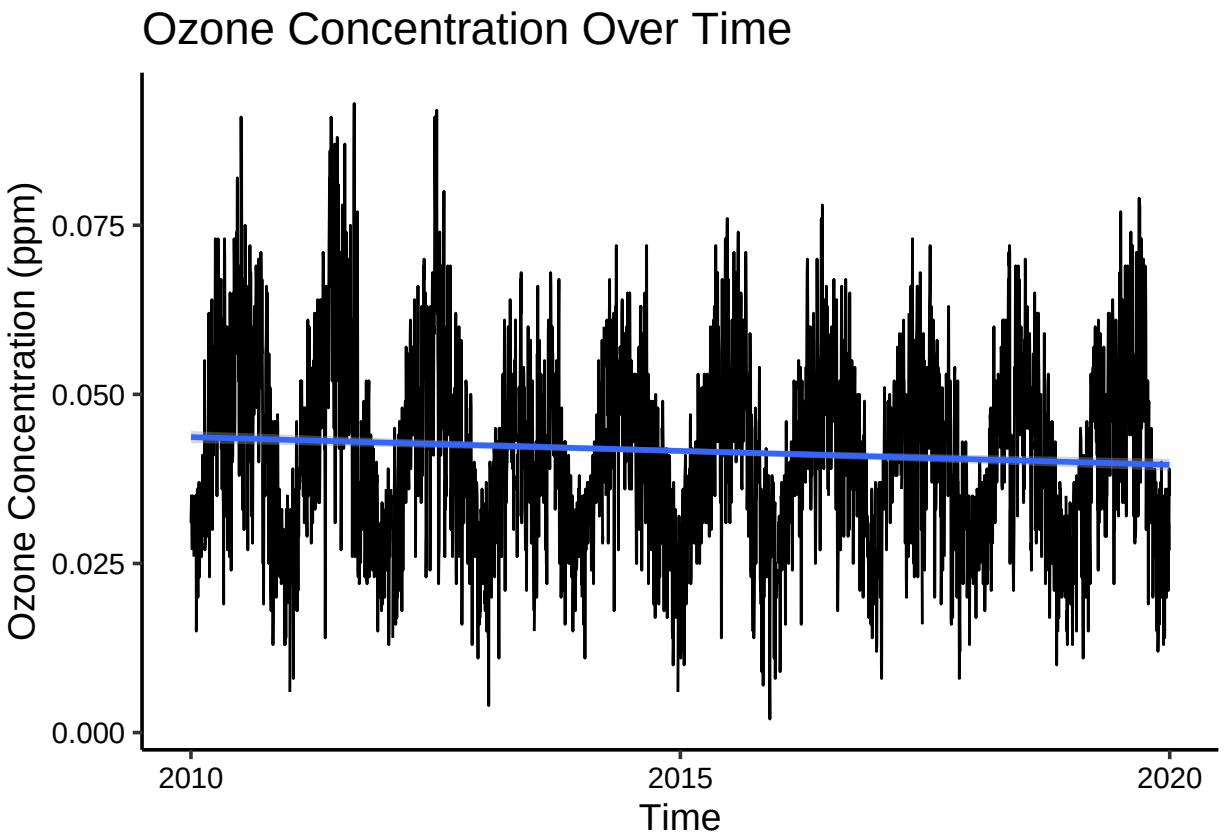
Visualize

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) +
  labs(title = "Ozone Concentration Over Time", x = "Time", y = "Ozone Concentration (ppm)")
```

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

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



Answer: This plot depicts a slight decrease in ozone concentration levels over time demonstrated by the downward sloping line of best fit.

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
head(GaringerOzone)
```

```
##         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 2010-01-06                                   NA              NA
```

```
summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

```
GaringerOzone <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

```
summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00200 0.03200 0.04100 0.04151 0.05100 0.09300
```

Answer:
We used linear because the piecewise constant uses the value of the nearest date which would not suit this data as we know it is seasonal and values fluctuate with time. While the spline interpolation is quadratic which would not be accurate to this data which we know is decreasing linearly.

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)

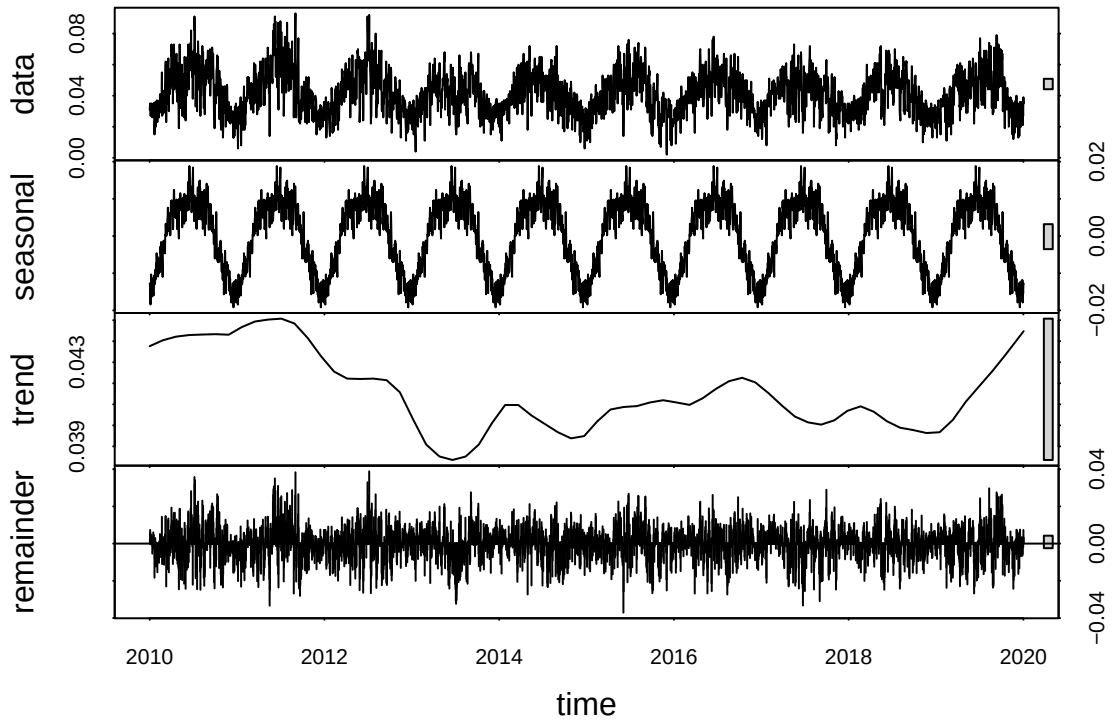
```
#9
GaringerOzone.monthly <-
  GaringerOzone %>%
  mutate(Month = month(Date), Year = year(Date))

GaringerOzone.monthly$Month.Year <-
  floor_date(GaringerOzone.monthly$Date, "month")

GaringerOzone.monthly <-
  GaringerOzone.monthly %>%
  group_by(Month.Year) %>%
  dplyr::summarize(OzoneConcentration =
                     mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  as.data.frame()
```

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.

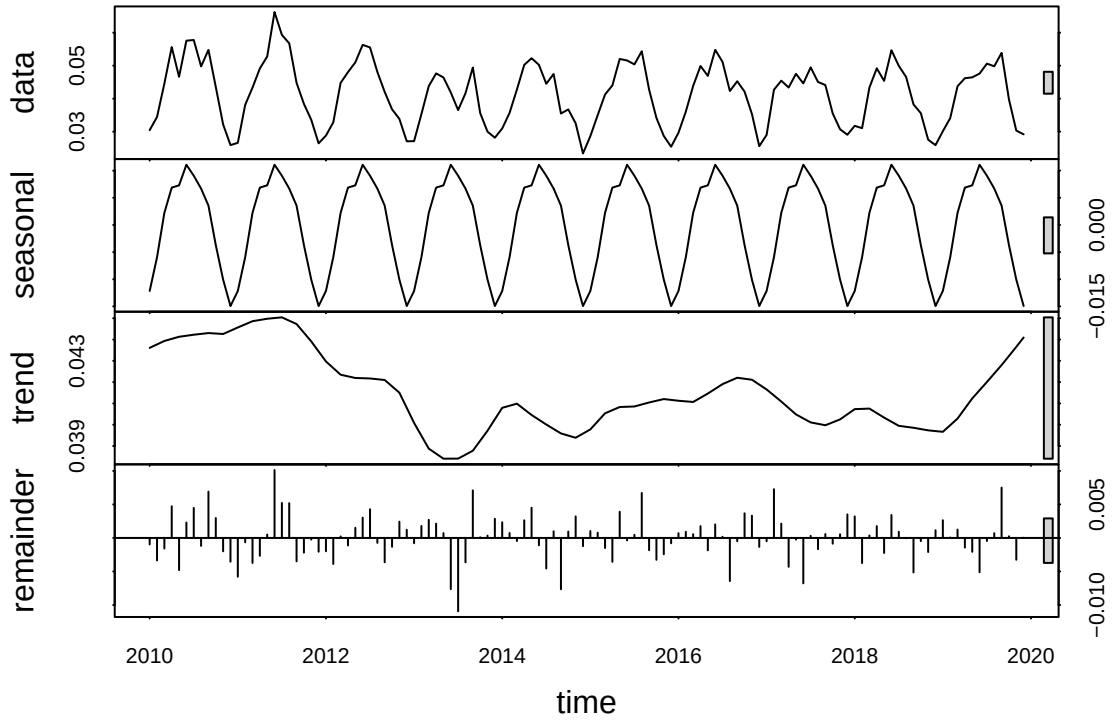
```
#10
f_month.daily <- month(first(GaringerOzone$Date))
f_year.daily <- year(first(GaringerOzone$Date))
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                   start=c(f_year.daily,f_month.daily),
                   frequency=365)

f_month.monthly <- month(first(GaringerOzone.monthly$Month.Year))
f_year.monthly <- year(first(GaringerOzone.monthly$Month.Year))
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$OzoneConcentration,
                   start=c(f_year.monthly,f_month.monthly),
                   frequency=12)
```

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

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




```
Monthly.Decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(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
Monthly.Trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
Monthly.Trend1
```

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

```
summary(Monthly.Trend1)
```

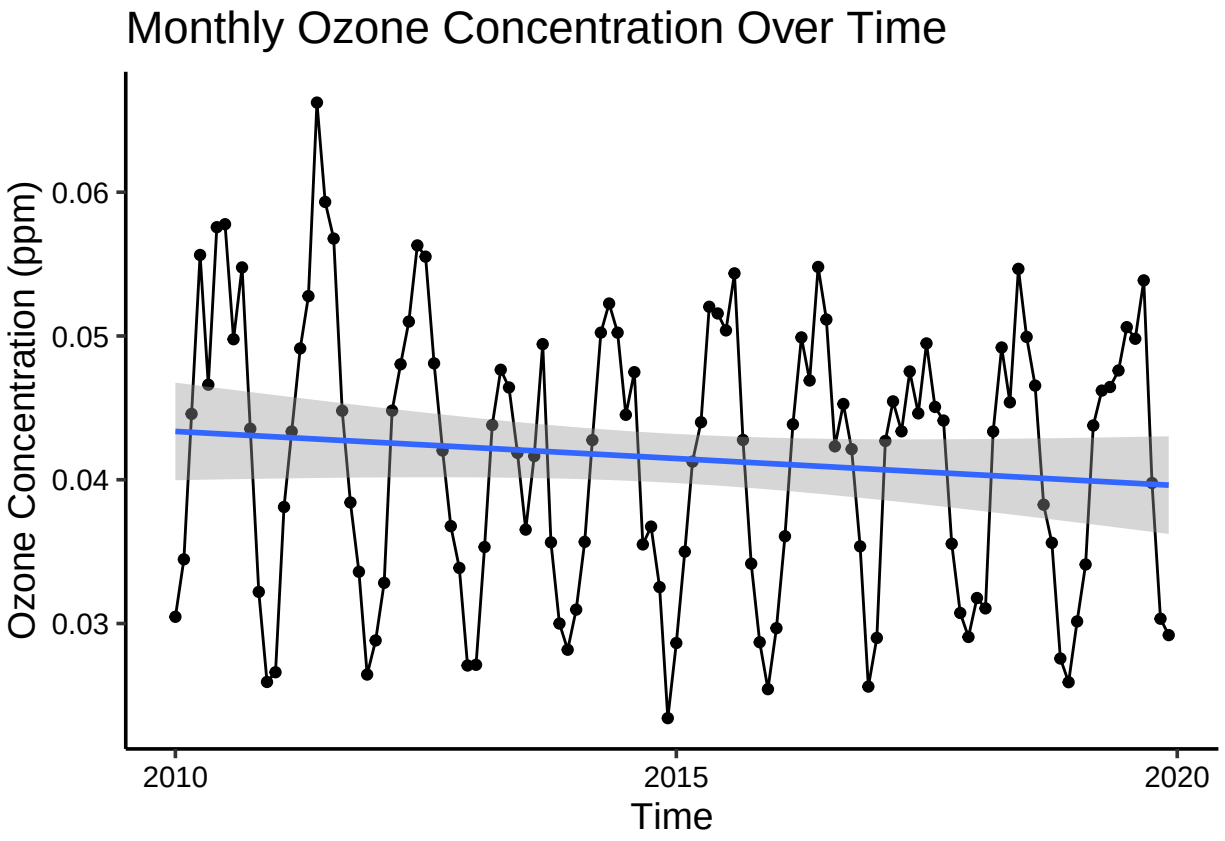
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer:
The seasonal Mann-Kendall allows for seasonality which is present in this data. It also is non-parametric and has no missin 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.

```
# 13
ggplot(GaringerOzone.monthly, aes(x = Month.Year, y = OzoneConcentration)) +
  geom_point() +
  geom_line() +
  geom_smooth(method = lm) +
  labs(title = "Monthly Ozone Concentration Over Time", x = "Time", y = "Ozone Concentration (ppm)")
```

```
## `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 ozone cnocentratin measurements have changed over the 2010s at this research station. The graph demonstrates that with room for statistical error there is a downward trajectory

of ozone concentrations. The results from the statistical test support this result with a p-value below 0.05 suggesting the results are not stationary (p = 0.0467).

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
Ozone.Components <- as.data.frame(Monthly.Decomp$time.series[,1:3])

Ozone.Components <- mutate(Ozone.Components,
        Observed = GaringerOzone.monthly$OzoneConcentration,
        Date = GaringerOzone.monthly$Month.Year)

f_month.monthly2 <- month(first(Ozone.Components$Date))
f_year.monthly2 <- year(first(Ozone.Components$Date))
GaringerOzone.monthly.nonseasonal.ts <- ts(Ozone.Components$trend,
                   start=c(f_year.monthly2,f_month.monthly2),
                   frequency=12)

#16
Monthly.Trend2 <- Kendall::MannKendall(GaringerOzone.monthly.nonseasonal.ts)
Monthly.Trend2
```

```
## tau = -0.269, 2-sided pvalue =1.3168e-05
```

```
summary(Monthly.Trend2)
```

```
## Score =  -1922 , Var(Score) = 194366.7
## denominator =  7140
## tau = -0.269, 2-sided pvalue =1.3168e-05
```

```
summary(Monthly.Trend1)
```

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

Answer: Without seasonality the Mann Kendal test is much more significant with a p-value of 1.32e-5. This is far below the significance cut off of 0.05 and below that of the seasonal test at 0.0467.