Assignment 7: Time Series Analysis

Jacob Freedman

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

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.

```
# 1
setwd("~/R/EDA-Fall2022")
getwd()
```

```
## [1] "/home/guest/R/EDA-Fall2022"
```

```
library(tidyverse)
library(lubridate)
library(trend)
library(zoo)
library(Kendall)
library(tseries)

theme1 <- theme_classic(base_size = 12) + theme(axis.text = element_text(color = "black"),
    legend.position = "left", axis.line = element_line(arrow = arrow()))

theme_set(theme1)

# 2
```

```r
EPAOzone2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
    stringsAsFactors = TRUE)
EPAOzone2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
    stringsAsFactors = TRUE)
EPAOzone2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
    stringsAsFactors = TRUE)
EPAOzone2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
    stringsAsFactors = TRUE)
EPAOzone2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
    stringsAsFactors = TRUE)
EPAOzone2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
    stringsAsFactors = TRUE)
EPAOzone2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
    stringsAsFactors = TRUE)
EPAOzone2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
    stringsAsFactors = TRUE)
EPAOzone2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
    stringsAsFactors = TRUE)
EPAOzone2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
    stringsAsFactors = TRUE)

GaringerOzone <- rbind(EPAOzone2010, EPAOzone2011, EPAOzone2012, EPAOzone2013, EPAOzone2014,
    EPAOzone2015, EPAOzone2016, EPAOzone2017, EPAOzone2018, EPAOzone2019)

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.

```r
# 3
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")
class(GaringerOzone$Date)
```

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

```r
# 4
GaringerOzoneProcessed <- select(GaringerOzone, 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"))
colnames(Days) <- c("Date")
```

```r
# 6
GaringerOzone <- left_join(Days, GaringerOzoneProcessed, by = c("Date"))
dim(GaringerOzone)
```

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

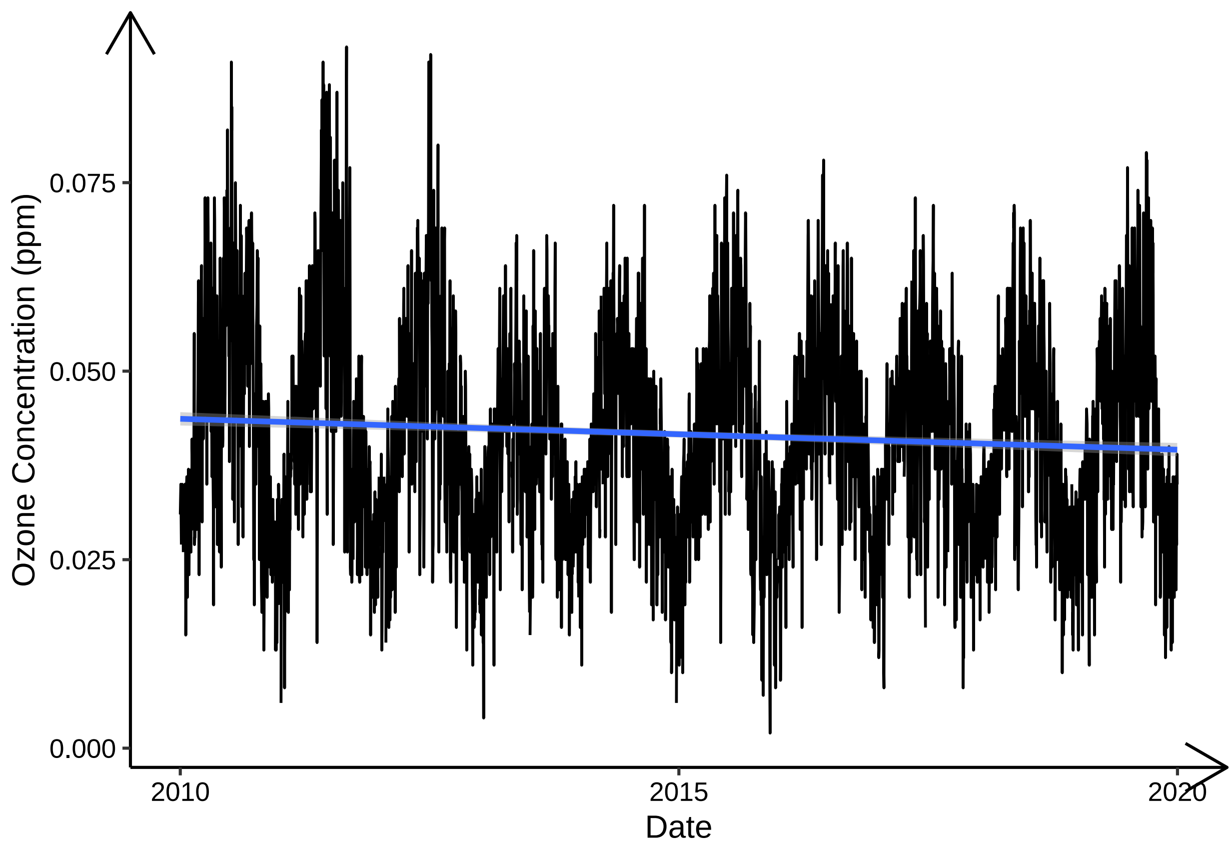
Visualize

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
OzonevTime <- ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
    geom_line() + geom_smooth(method = lm) + ylab("Ozone Concentration (ppm)") +
    theme1
print(OzonevTime)
```

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

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



Answer: There appears to be a seasonal trend in ozone concentration that repeats annually. Overall, from 2010-2019, ozone concentrations appear to be in gradual decline.

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

summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)  #63 NAs
```

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

```
summary(GaringerOzone_clean$Ozone.clean)  #No NAs
```

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

Answer: Piecewise may not work because a missing observation would be equidistant to measurements one day before and one day after, so it would not have a single nearest neighbor. Spline may not be the best method because we do not know enough about our data to assume that a polynomial based interpolation is superior to linear.

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_clean %>%
    mutate(Month = month(Date)) %>%
    mutate(Year = year(Date)) %>%
    group_by(Year, Month) %>%
    summarize(MeanMonthlyOzone = mean(Ozone.clean)) %>%
    mutate(Month_Year = my(paste(Month, "-", Year)))
```

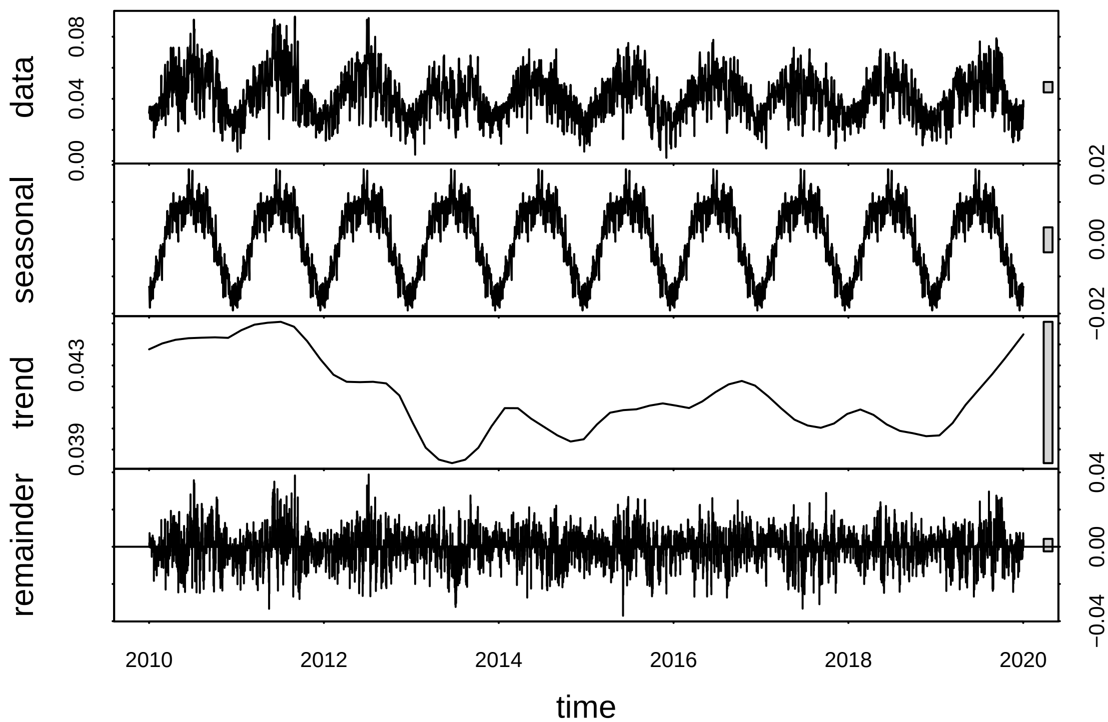
```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

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.

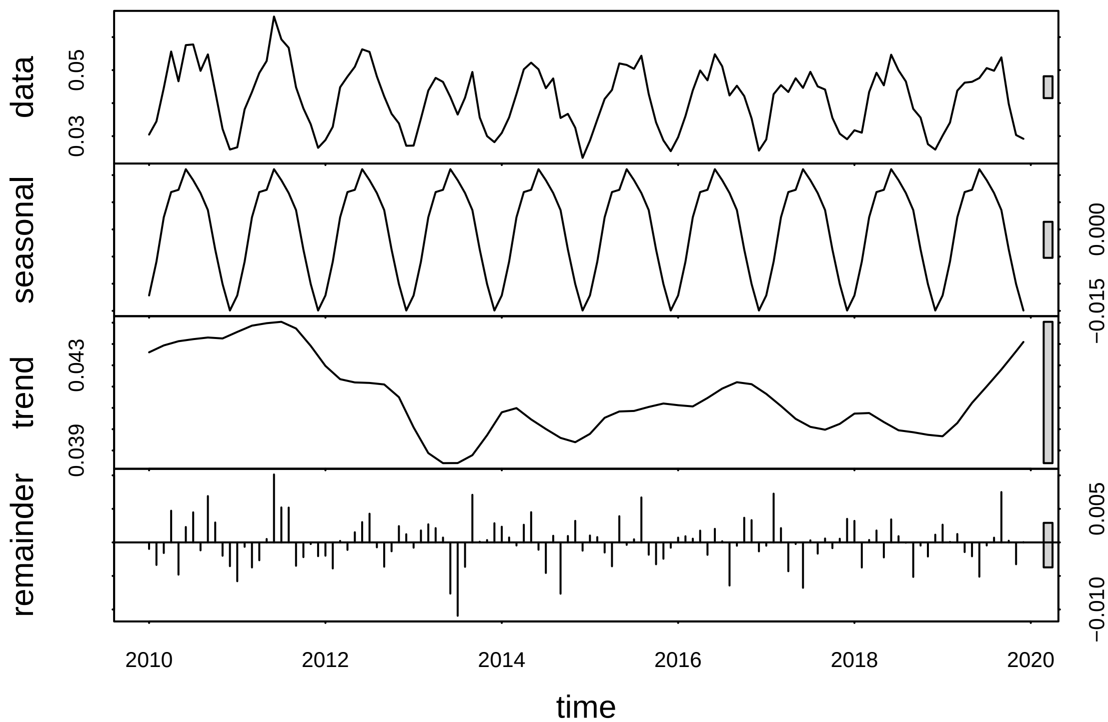
```
# 10
GaringerOzone.daily.ts <- ts(GaringerOzone_clean$Ozone.clean, start = c(2010, 1),
    end = c(2019, 367), frequency = 365)
# This end function puts the time series at the correct final day (12/31/2019)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$MeanMonthlyOzone, start = c(2010,
    1), end = c(2019, 12), frequency = 12)
```

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

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




```r
Ozone.monthly.decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(Ozone.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
Ozonetrend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(Ozonetrend)
```

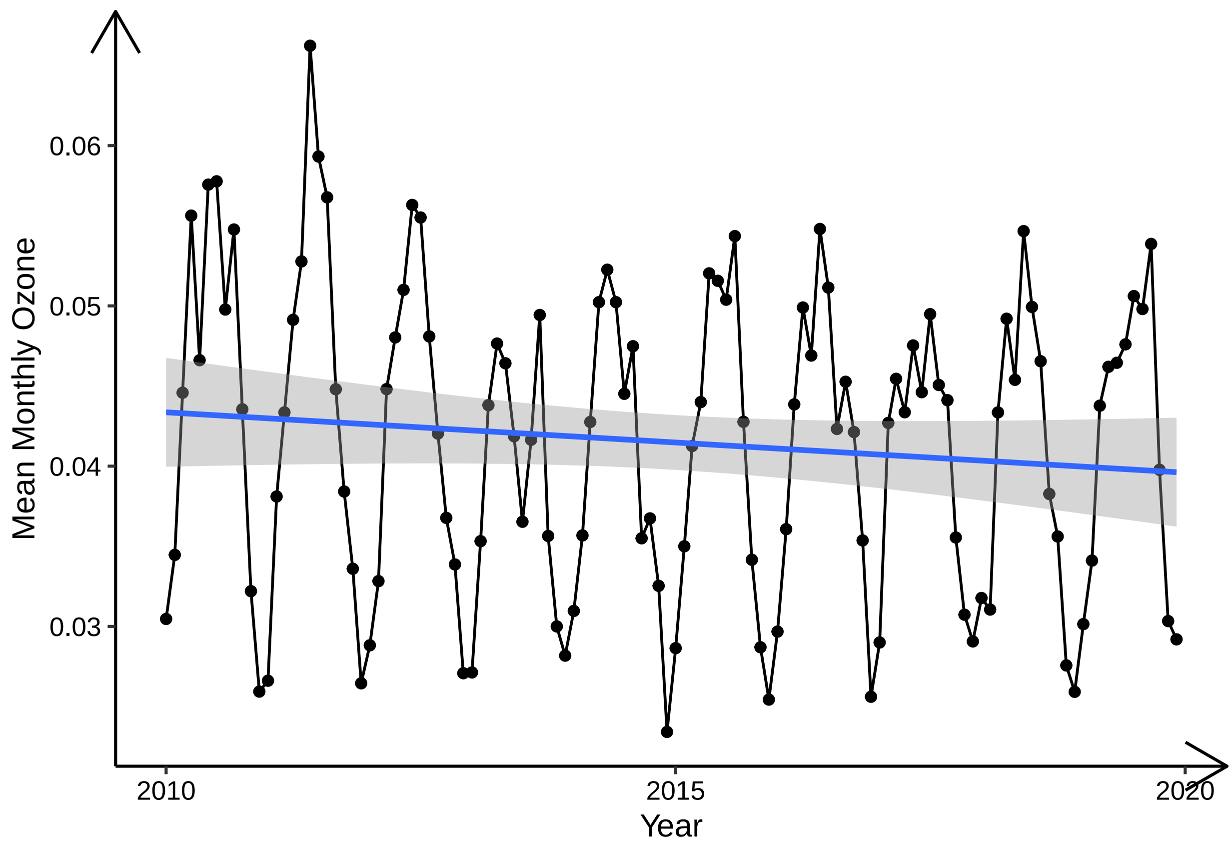
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The seasonal Mann Kendall test is most appropriate because the data appears to be cyclical annually and does not appear to be stationary. If the data was not seasonal we could use an ADF test.

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 Also added a linear regression line to visualize ozone concentration
# decline.
MonthlyOzonePlot <- ggplot(GaringerOzone.monthly, aes(x = Month_Year, y = MeanMonthlyOzone)) +
    geom_point() + geom_line() + geom_smooth(method = lm) + ylab("Mean Monthly Ozone") +
    xlab("Year") + theme1
print(MonthlyOzonePlot)
```

```
## `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: Ozone concentrations at Garinger High School decreased significantly between 2010 and 2020 despite annual fluctuation (Seasonal Mann Kendall - p = 0.047). This is supported by the "trend" line in the monthly decomposition plot, which is generally declining (with the exception of 2019, possibly because most of the 2019 data is upward-trending due to seasonality).

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_Components <- as.data.frame(Ozone.monthly.decomp$time.series[, 1:3]) %>%
    mutate(NonSeasonality = trend + remainder)

# 16
GaringerOzone.monthly.ts2 <- ts(GaringerOzone_Components$NonSeasonality, start = c(2010,
    1), end = c(2019, 12), frequency = 12)

MannKendall(GaringerOzone.monthly.ts2)
```

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

Answer: The Mann-Kendall Test showed that ozone concentrations decreased significantly despite extracting seasonality (p = 0.0075). The Mann-Kendall Test on extracted data showed more significant results than the Seasonal Mann Kendall on the original data.