Assignment 7: Time Series Analysis

Reino Hyyppa

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 working directory
getwd()
```

```
## [1] "/Users/reinohyyppa/Desktop/Duke MEM/Spring 22 /ENV872/Environmental_Data_Analytics_2022"
```

```
# 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   2.0.2     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("dplyr")
library("readr")

# create ggplot theme
ggtheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(ggtheme) # 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.

```
# initalize data
d1 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010
d2 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011
d3 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012
d4 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013
d5 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014
d6 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015
d7 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016
d8 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017
d9 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018
d10 <- read.csv("../Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC201

# combine data into single dataframe
GaringerOzone <- rbind(d1,d2,d3,d4,d5,d6,d7,d8,d9,d10)
```

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.

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

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

# create new dates data frame
sd <- as.Date("2010-01-01")
ed <- as.Date("2019-12-31")
days <- as.data.frame(seq.Date(sd, ed, by = 1))
colnames(days) <- "Date"

# combine data frames
GaringerOzone <- left_join(days, GaringerOzone)
```

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

Visualize

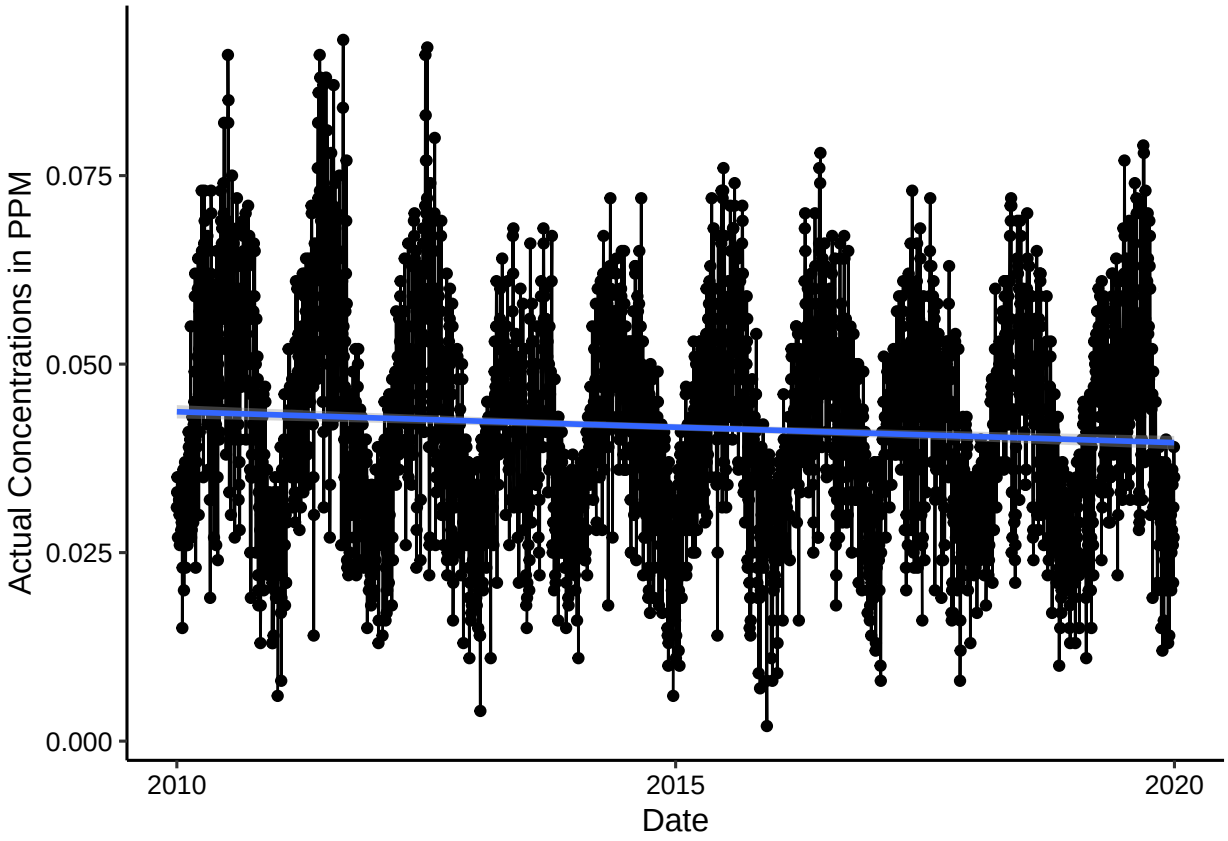
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
# ggplot of concentrations over time
ozone_data_plot <-
  ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_point() +
  geom_line() +
  ylab("Actual Concentrations in PPM") +
  geom_smooth(method = "lm")
print(ozone_data_plot)
```

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

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

```
## Warning: Removed 63 rows containing missing values (geom_point).
```



Answer: Yes, based on the plot of ozone concentrations, there appears to be a slight decline in 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?

```
# linear interpolation to fill in NAs
GaringerOzone_clean <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

# check for NAs
summary(GaringerOzone_clean$Daily.Max.8.hour.Ozone.Concentration) # no NAs
```

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

Answer: We used the linear interpolation method to fill in NAs in the ozone concentration data because we know that values are equally spaced (1 day interval). A piecewise constant and spline interpolation would not be appropriate for this dataset because we are dealing with seasonal data that fluctuates at regular intervals.

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)

```
# new data frame over ozone averages by month
GaringerOzone.monthly <-
 GaringerOzone_clean %>%
  mutate(Month = month(Date),
         Year = year(Date)) %>%
  mutate(Date = my(paste0(Month, "-", Year))) %>%
  dplyr::group_by(Date, Month, Year) %>%
  dplyr::summarise(mean_Ozone = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  select(mean_Ozone, Date)
```

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

```
## Adding missing grouping variables: `Month`
```

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.

```
# time series of daily observations
GaringerOzone.daily.ts <- ts(GaringerOzone_clean$Daily.Max.8.hour.Ozone.Concentration, start = c(2010,1)
head(GaringerOzone.daily.ts)
```

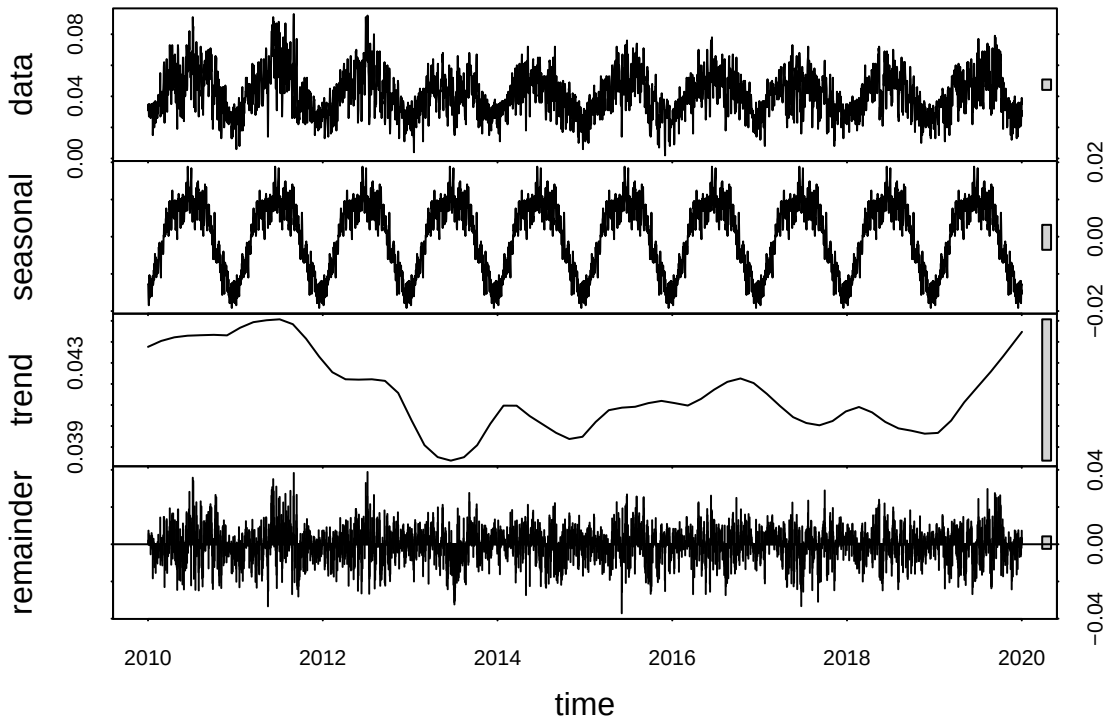
```
## [1] 0.031 0.033 0.035 0.031 0.027 0.030
```

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

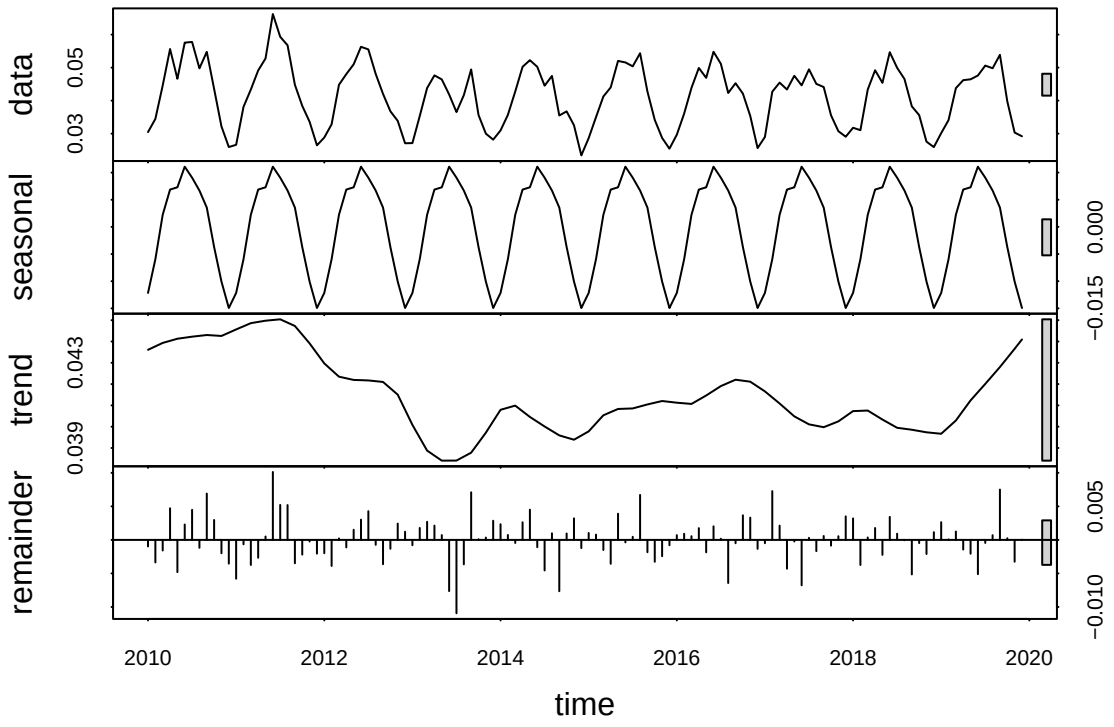
```
## [1] 0.03046774 0.03446429 0.04458065 0.05563333 0.04661290 0.05756667
```

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

```
# decompose daily time series
GaringerOzone.daily_Decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily_Decomposed)
```




```
# decompose monthly time series
GaringerOzone.monthly_Decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(GaringerOzone.monthly_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?

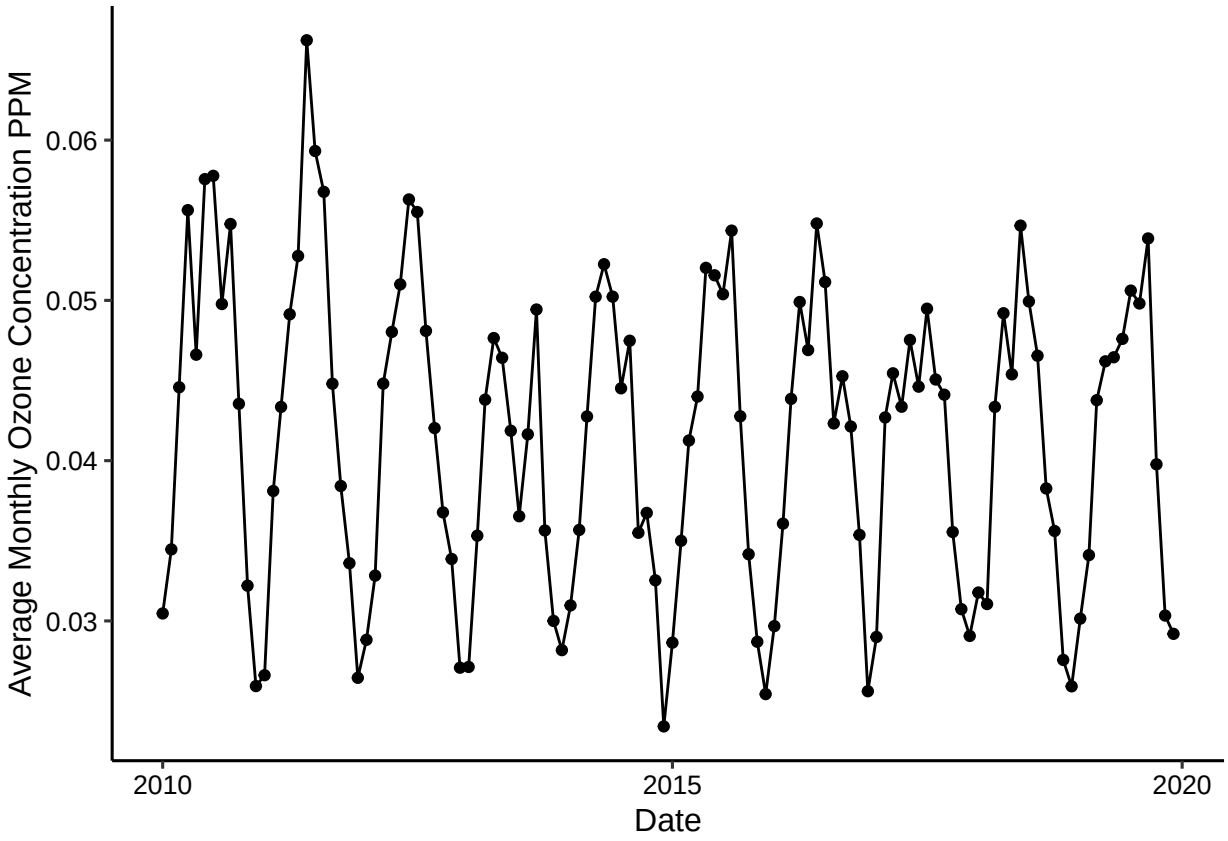
```
# Run seasonal MK test
ozone_data_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Inspect results
ozone_data_trend1
```

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

Answer: The Mann-Kendall is best suited for environmental and climatic data that is non-stationary, in other words this test can be performed on all types of distributions. More specifically, the seasonal Mann-Kendall tests for monotonic trends in seasonal data, that is data collected over periods of time in which upwards or downward trends are observed.

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.

```
# ggplot of monthly ozone concentration totals
ozone_monthly_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_Ozone)) +
  geom_point() +
  geom_line() +
  ylab("Average Monthly Ozone Concentration PPM")
print(ozone_monthly_plot)
```



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 results of the Seasonal Mann-Kendall test conclude that there is a monotonic trend in the ozone concentrations during the 2010s with a negative tau value of -0.143. The p-value of 0.047 is below the alpha (<0.05) and, therefore, we reject the null hypothesis that there is no monotonic trend in the time series data. Thus, there is a statistically significant decline in Ozone concentrations during the 2010s. (tau = -0.143, 2-sided pvalue =0.046724)

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.

```
# remove seasonal component from time series
GaringerOzone.monthly_Components <- as.data.frame(GaringerOzone.monthly_Decomposed$time.series[,1:3])

GaringerOzone.monthly_Components <-
  mutate(GaringerOzone.monthly_Components,
         Observed = GaringerOzone.monthly$mean_Ozone,
         Date = GaringerOzone.monthly$Date)

# make new time series using non-seasonal Ozone monthly totals
```

```
fmonth <- month(first(GaringerOzone.monthly_Components$Date))
fyear <- year(first(GaringerOzone.monthly_Components$Date))
GaringerOzone.mo.ts2 <- ts(GaringerOzone.monthly_Components$Observed, start = c(fmonth, fyear), frequen

# run the Mann-Kendall test on non-seasonal series
Ozone_trend2 <- Kendall::MannKendall(GaringerOzone.mo.ts2)
Ozone_trend2
```

```
## tau = -0.0594, 2-sided pvalue =0.33732
```

Answer: After removing the seasonal component of the Garinger Ozone monthly time series, and rerunning the time series, there is no longer a significant monotonic trend in the ozone concentration dataset. The Mann-Kendall on the non-seasonal Ozone data revealed a small, negative tau of -0.0594. The p-value for the non-seasonal data increased from 0.047 to 0.337 and, therefore, we fail to reject the null hypothesis that there is no monotonic trend in the time series data. (tau = -0.0594, 2-sided pvalue =0.33732)