Assignment 8: Time Series Analysis

Cristiana Falvo

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., "Salk_A06_GLMs_Week1.Rmd") prior to submission.

The completed exercise is due on Tuesday, March 3 at 1:00 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme
- 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). Call these GaringerOzone201*, with the star filled in with the appropriate year in each of ten cases.

```
getwd()
```

```
## [1] "/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analytics_2020"
```

```
library(tidyverse)
library(lubridate)
library(zoo)
library(trend)
library(scales)

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

# data
GraingerOzone2010 <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analyti
GraingerOzone2011 <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analyti
GraingerOzone2012 <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analyti
GraingerOzone2013 <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analyti
```

```
GraingerOzone2014 <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analyti
GraingerOzone2015 <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analyti
GraingerOzone2016 <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analyti
GraingerOzone2017 <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analyti
GraingerOzone2018 <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analyti
GraingerOzone2019 <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analyti
```

Wrangle

2. Combine your ten datasets into one dataset called GaringerOzone. Think about whether you should use a join or a row bind.
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 2019-12-13 (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 comine 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.

```
# 2
dim(GraingerOzone2010)
```

```
## [1] 360  20
```

```
dim(GraingerOzone2011)
```

```
## [1] 359  20
```

```
colnames(GraingerOzone2010)
```

```
##  [1] "Date"
##  [2] "Source"
##  [3] "Site.ID"
##  [4] "POC"
##  [5] "Daily.Max.8.hour.Ozone.Concentration"
##  [6] "UNITS"
##  [7] "DAILY_AQI_VALUE"
##  [8] "Site.Name"
##  [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
```

```r
colnames(GraingerOzone2011)
```

```
##  [1] "Date"
##  [2] "Source"
##  [3] "Site.ID"
##  [4] "POC"
##  [5] "Daily.Max.8.hour.Ozone.Concentration"
##  [6] "UNITS"
##  [7] "DAILY_AQI_VALUE"
##  [8] "Site.Name"
##  [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
```

```r
dim(GraingerOzone2012)
```

```
## [1] 359  20
```

```r
dim(GraingerOzone2015)
```

```
## [1] 364  20
```

```r
dim(GraingerOzone2016)
```

```
## [1] 359  20
```

```r
dim(GraingerOzone2018)
```

```
## [1] 354  20
```

```r
359 * 10
```

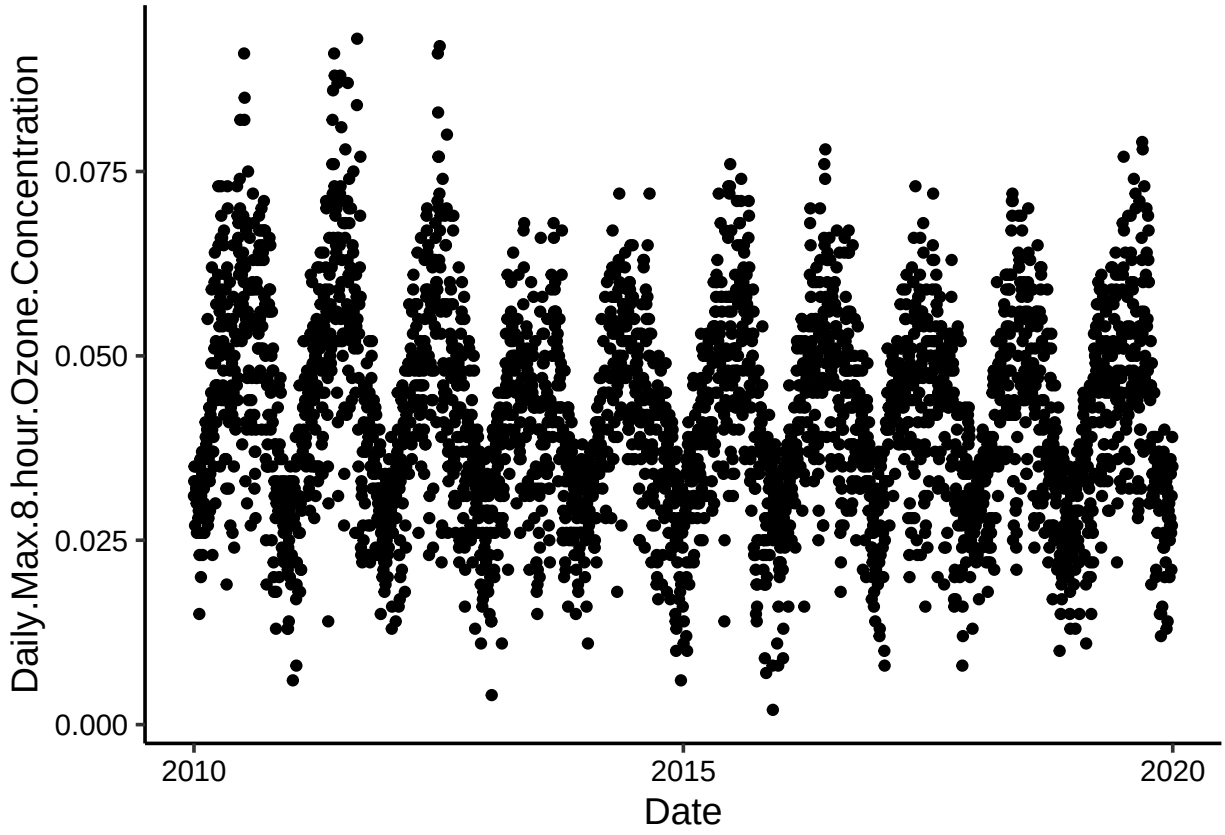
```
## [1] 3590
```

```r
# combine two tables (test)
table_2010_11 <- rbind(GraingerOzone2010, GraingerOzone2011)

Combined <- rbind(GraingerOzone2010, GraingerOzone2011, GraingerOzone2012, GraingerOzone2013, GraingerO

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

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

```r
# 4
# ** what's a good way to see if we have NA values in date column
  # do we have every day, with some days having NA as the ozone value
  # or do we have missing days (how do we spit this out)
Combined_subset <- select(Combined, Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
```

```
combo_subset_plot <-
  ggplot(Combined_subset, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_point()
print(combo_subset_plot)
```




```
str(Combined_subset)
```

```
## 'data.frame':    3589 obs. of  3 variables:
##  $ Date                                : Date, format: "2010-01-01" "2010-01-02" ...
##  $ Daily.Max.8.hour.Ozone.Concentration: num  0.031 0.033 0.035 0.031 0.027 0.033 0.035 0.032 0.032 0
##  $ DAILY_AQI_VALUE                     : int  29 31 32 29 25 31 32 30 30 28 ...
```

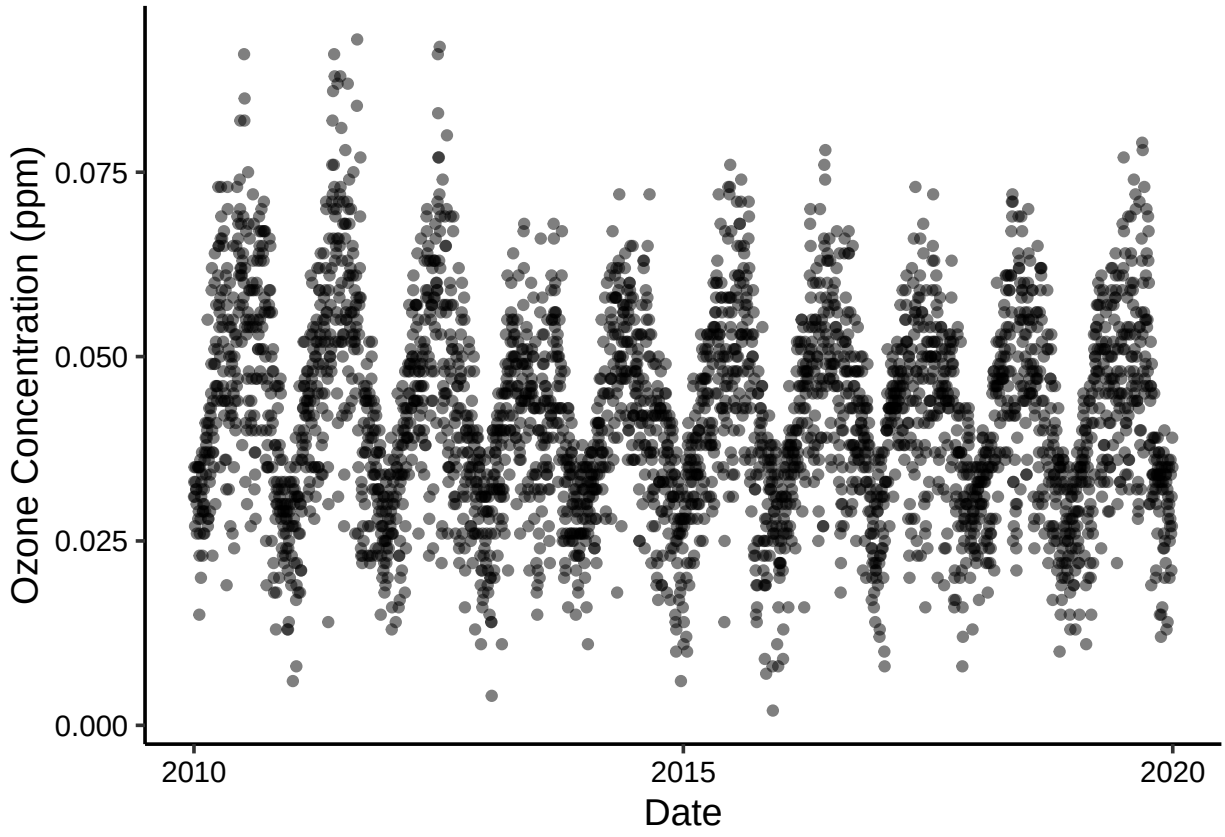
```
# 5

Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), "days"))

# colnames(Days)
# colnames(Days) <- colnames()
names(Days)[1] <- "Date"
# colnames(Days)
# # 6
GraingerOzone <- left_join(Days, Combined_subset %>%
                             select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE),
                           by = c("Date" = "Date"))
```

Visualize

7. Create a ggplot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly.

```
Ozone_ot <-
   ggplot(GraingerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
   geom_point(alpha = .5) +
   labs(x = "Date", y = "Ozone Concentration (ppm)")

 print(Ozone_ot)
```

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



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?

   Answer: We used a linear interpolation in order to take both past and future data into account when predicting in between values (this is opposed to using piecewise constant interpolation, which would make the missing data the same as the closest point on either side). We chose linear over spline because the spline method would potentially give us a predicted value that's too specific for the extrapolation we're making.

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) # # 10. Generate a time series called GaringerOzone.monthly.ts, with a monthly frequency that specifies the correct start and end dates. # # 11. Run a time series analysis. In this case the seasonal Mann-Kendall is most appropriate; why is this? # > Answer: Seasonal Mann-Kendall is the most appropriate time series analysis to run because we're looking for a longterm monotonic trend in data that was collected year-round, is not normally distributed and is not temporally autocorrelated.

10. To figure out the slope of the trend, run the function `sea.sens.slope` on the time series dataset.

11. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. No need to add a line for the seasonal Sen's slope; this is difficult to apply to a graph with time as the x axis. Edit your axis labels accordingly.

```
# 8 interpolate missing data (linear interp method)
GraingerOzone$Daily.Max.8.hour.Ozone.Concentration <- na.approx(GraingerOzone$Daily.Max.8.hour.Ozone.Co

# 9 organize into monthly data
GraingerOzone_monthly <-
  GraingerOzone %>%
  mutate(Year = year(Date),
         Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarise(MonthlyMeanOzone = mean(Daily.Max.8.hour.Ozone.Concentration))

GraingerOzone_monthly$Date <- as.Date(paste(GraingerOzone_monthly$Year,
                                            GraingerOzone_monthly$Month,
                                            1, sep = "-"),
                                      format = "%Y-%m-%d")
# 10 create time series
GaringerOzone.monthly.ts <- ts(GraingerOzone_monthly$MonthlyMeanOzone, frequency = 12,
                               start = c(2010, 1, 1), end = c(2019, 12, 31))

# 11
  # high positive S = increasing trend
  # low negative S = decreasing trend
GaringerOzone.monthly.trend <- smk.test(GaringerOzone.monthly.ts)
GaringerOzone.monthly.trend
```

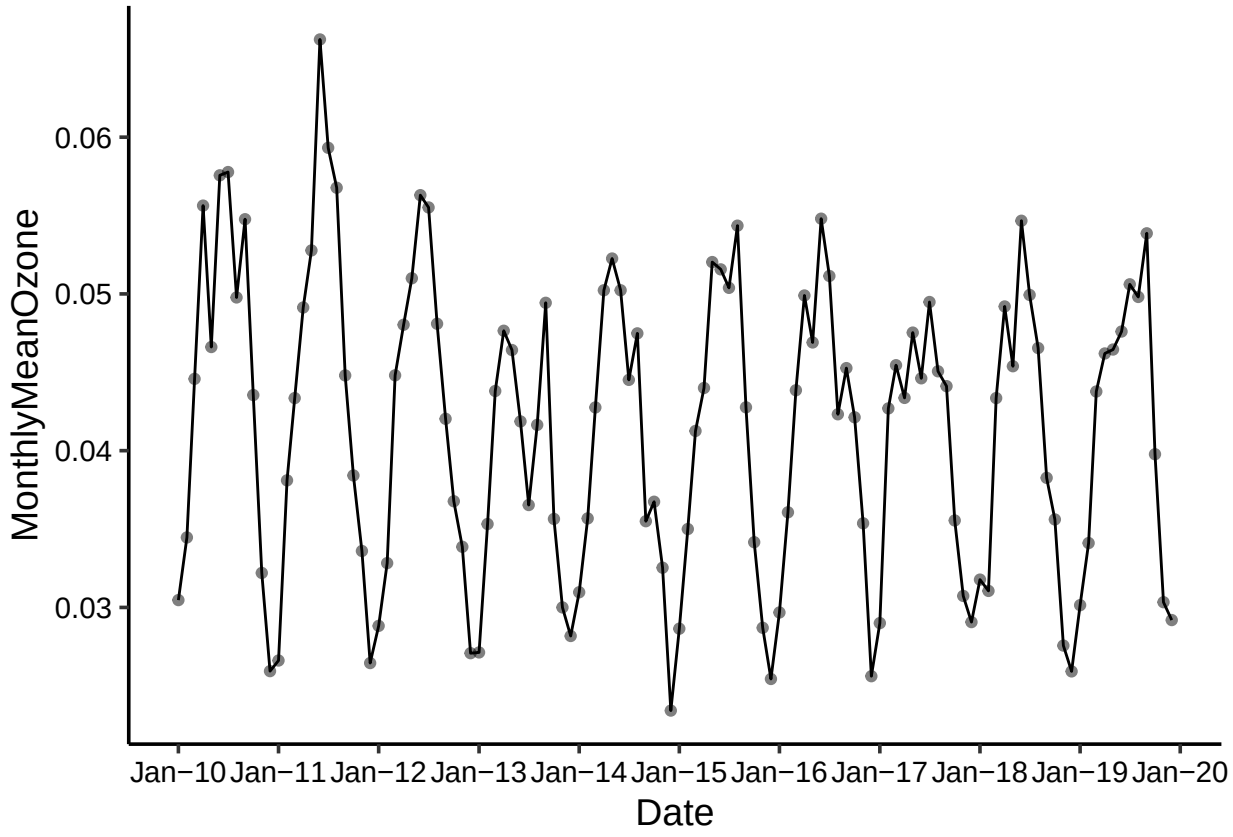
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
```

```
##    S varS
##  -77 1499
```

```
# 12
sea.sens.slope(GaringerOzone.monthly.ts)
```

```
## [1] -0.0002044163
```

```
# 13
Ozone_ot_mean.monthly <-
  ggplot(GraingerOzone_monthly, aes(x = Date, y = MonthlyMeanOzone)) +
  geom_point(alpha = .5) +
  scale_x_date(date_breaks = "year" , date_labels = "%b-%y") +
  geom_line()

print(Ozone_ot_mean.monthly)
```



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 monthly mean ozone values from Garinger High School have consitent seasonal spikes during the summer months (as shown by the graph) and an overall decreasing monotonic trend (as suggested by our Mann-Kendall test). The overall decreasing trend in ozone concentration is slight, as reflected by our slope calculation (Seasonal Mann-Kendall trend test: S = -77, sea sens slope test: slope = -0.0002).