Assignment 7: Time Series Analysis

Lydie Costes

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. Get wd, load packages, and set theme
getwd()
```

```
## [1] "/Users/lydiecostes/Documents/Duke/DataAnalytics/GithubRepos/Environmental_Data_Analytics_2022/As
```

```
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(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
```

```
require(data.table)
```

```
## Loading required package: data.table
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

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

```
#install.packages("chron")
library(chron)
```

```
##
## Attaching package: 'chron'

## The following object is masked from 'package:tseries':
##
##     is.weekend

## The following objects are masked from 'package:lubridate':
##
##     days, hours, minutes, seconds, years
```

```
theme_set(theme_bw())
```

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. Import files

# set working directory to the folder with the data
setwd("../Data/Raw/Ozone_TimeSeries/")
# create a list of files
ozone_files <- list.files(pattern = "*.csv")
# combine files into one dataset
ozone_dataset = do.call(rbind, lapply(ozone_files, fread))
# remove intermediary file
rm(ozone_files)
# make dataset a dataframe
GaringerOzone <- as.data.frame(unclass(ozone_dataset))

setwd("../")
```

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, "%m/%d/%Y")
```

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

# 5.
# List all dates
Days <- as.data.frame(seq(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"), by = "days"))
# Rename column
colnames(Days) <- "Date"

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

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

```
colnames(GaringerOzone)[2] <- "O3ConcMax"
```

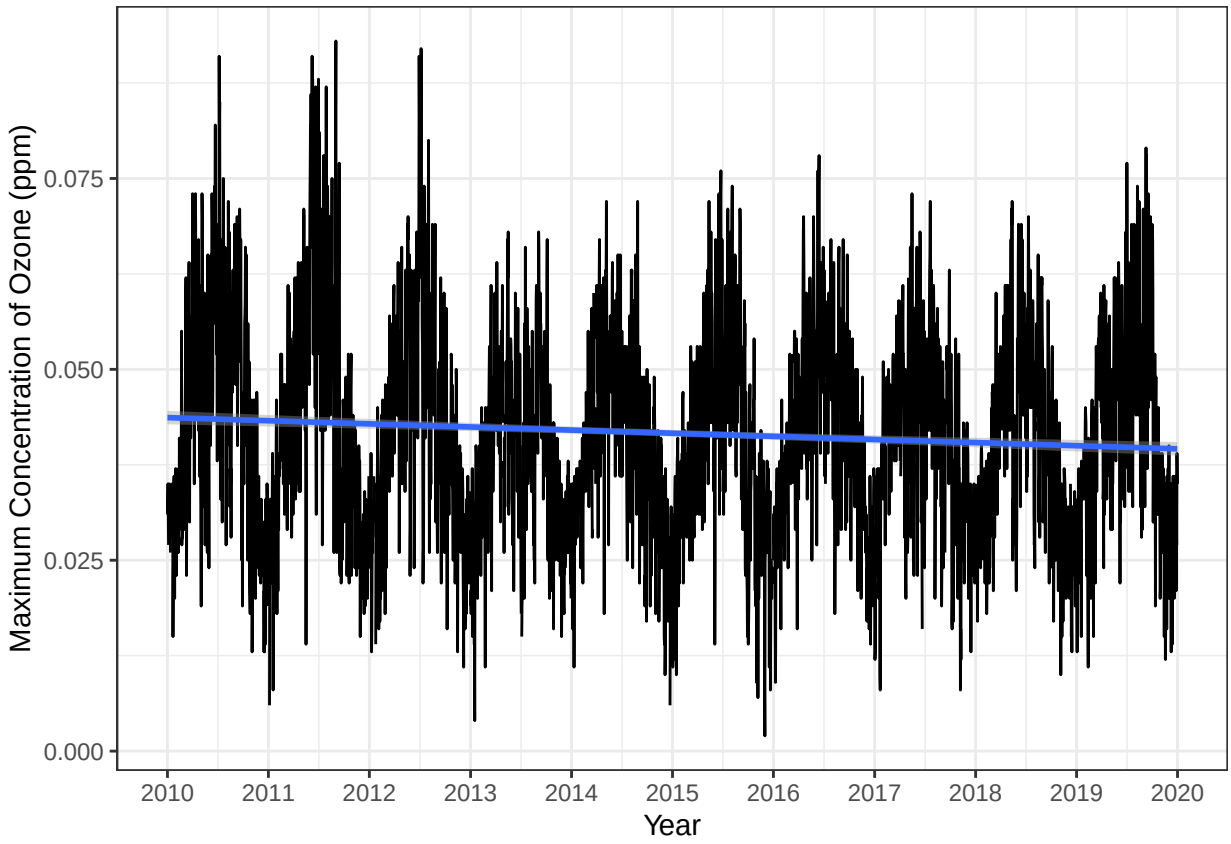
Visualize

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
ggplot(GaringerOzone, aes(x=Date, y=O3ConcMax)) +
  geom_line() +
  geom_smooth(method="lm") +
  labs(x="Year", y="Maximum Concentration of Ozone (ppm)") +
  scale_x_date(date_breaks = "years", date_labels="%Y")
```

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

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



Answer: There is a slight downward trend in ozone levels over time, but we would need to run statistical analyses to see whether the trend is significant.

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(O3ConcMax.clean = zoo::na.approx(O3ConcMax))
```

Answer: We don't use the piecewise constant method because we don't have any reason to believe that the date immediately before or immediately after the missing date is a better representation of that day. We don't use the spline method because we don't have a reason to expect that the pattern is quadratic.

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(Year = year(Date),
         Month = month(Date)) %>%
  group_by(Year, Month) %>%
  drop_na() %>%
  summarise(Average_O3 = mean(O3ConcMax)) %>%
  mutate(Date = as.Date(paste(Month, "01", Year, sep="/"), "%m/%d/%Y"))
```

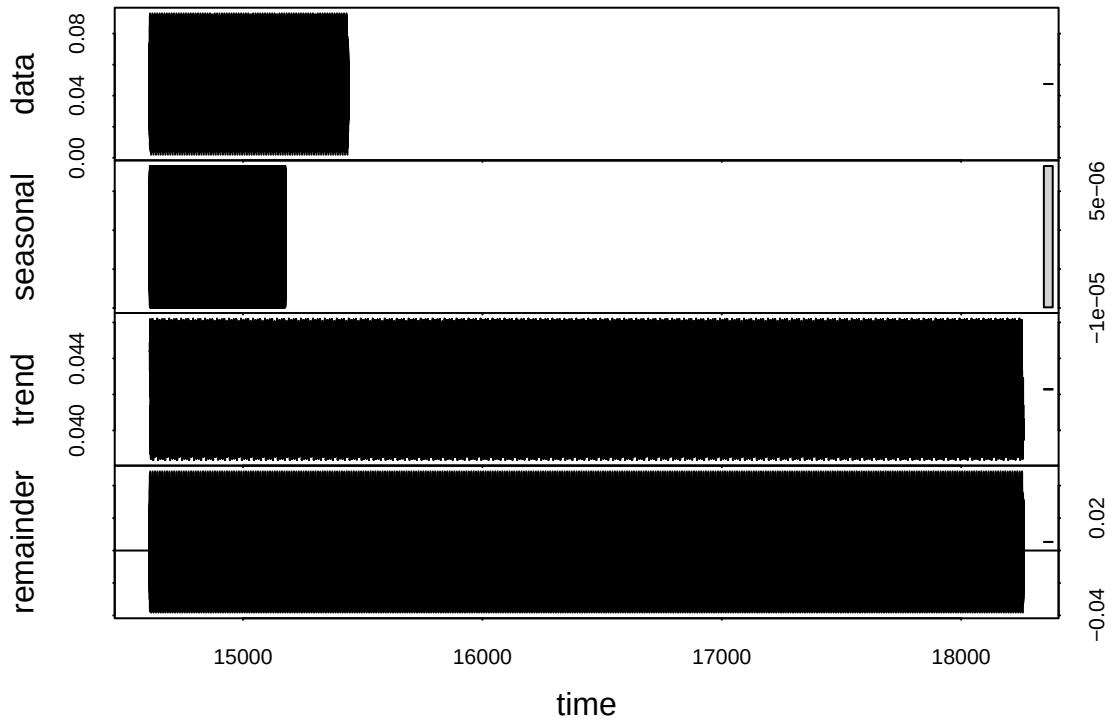
```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.
```

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.

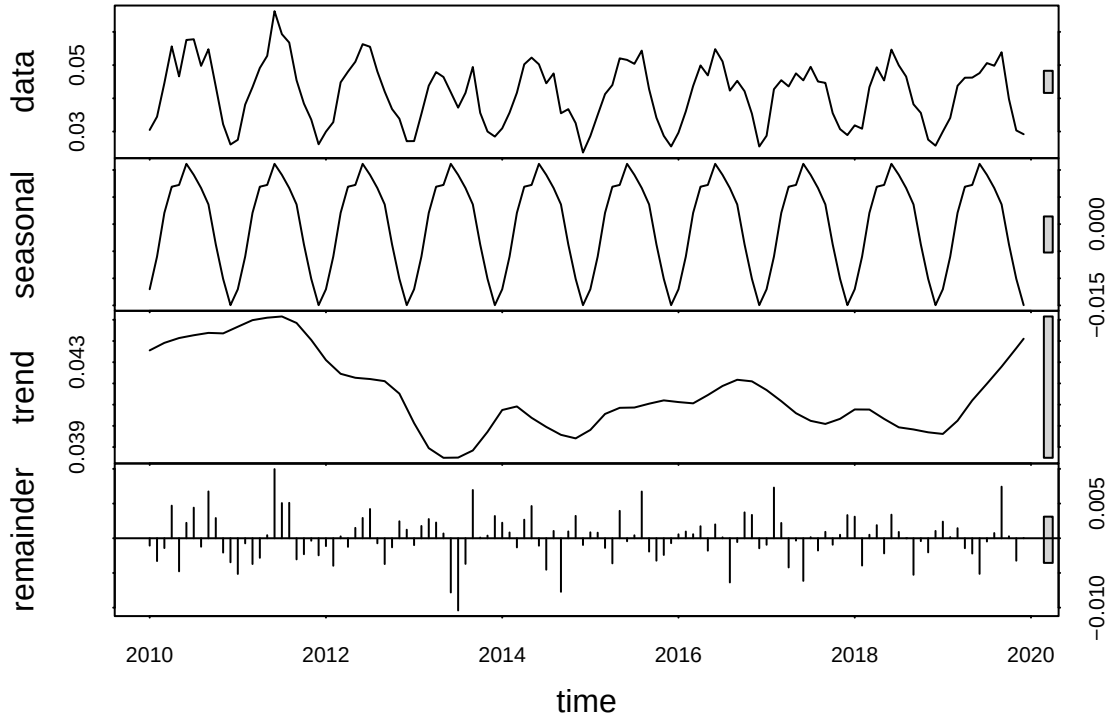
```
#10
# Pull start and end dates
fdate <- date(first(GaringerOzone$Date))
ldate <- date(last(GaringerOzone$Date))
GaringerOzone.daily.ts <- ts(GaringerOzone_clean$O3ConcMax.clean,
                             start = fdate,
                             end = ldate,
                             frequency = 365)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Average_O3,
                               start = c(year(fdate), month(fdate)),
                               end = c(year(ldate), month(ldate)),
                               frequency = 12)
```

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

```
#11
# Decompose and plot daily data
GaringerOzone.daily.ts.decomposed <- stl(GaringerOzone.daily.ts,
                                         s.window = "periodic")
plot(GaringerOzone.daily.ts.decomposed)
```




```
# Decompose and plot monthly data
GaringerOzone.monthly.ts.decomposed <- stl(GaringerOzone.monthly.ts,
                                           s.window = "periodic")
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
Garinger.trend <- SeasonalMannKendall(GaringerOzone.monthly.ts)

summary(Garinger.trend)
```

```
## Score =  -88 , Var(Score) = 1498
## denominator =  538.9944
## tau = -0.163, 2-sided pvalue =0.022986
```

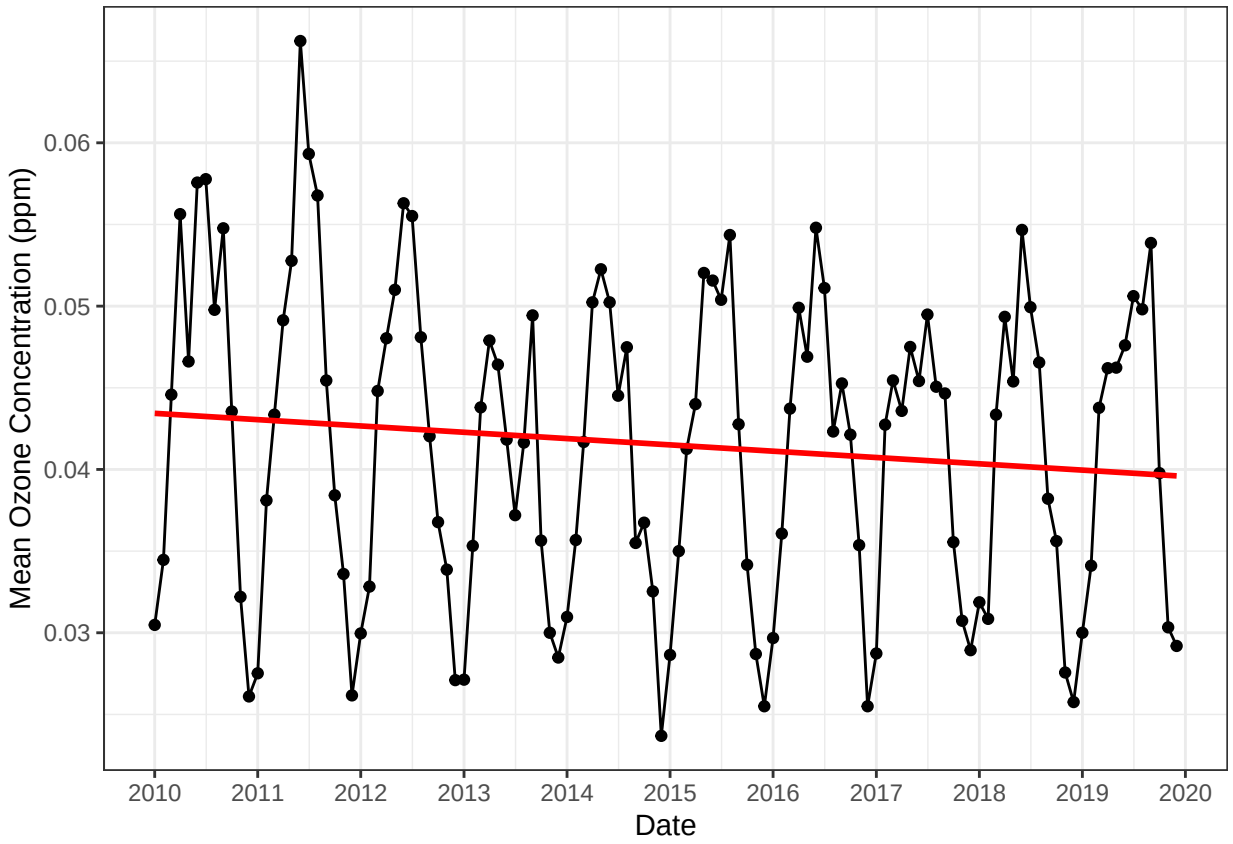
```r
# Because the p-value is less than 0.05, we can reject the null. There does
# appear to be a trend in the data.
```

Answer: The Seasonal Mann-Kendall method is most appropriate because we can see seasonal trends over each year, and this method is the only one that takes into account seasonality.

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
ggplot(GaringerOzone.monthly, aes(Date, Average_O3)) +
  geom_point() +
  geom_line() +
  labs(x = "Date", y = "Mean Ozone Concentration (ppm)") +
  scale_x_date(date_breaks = "years", date_labels="%Y") +
  geom_smooth(method = "lm", color = "red", se = FALSE)
```

```
## `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 have decreased over the 2010s at this station (tau = -0.163, 2-sided p-value = 0.023).

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.

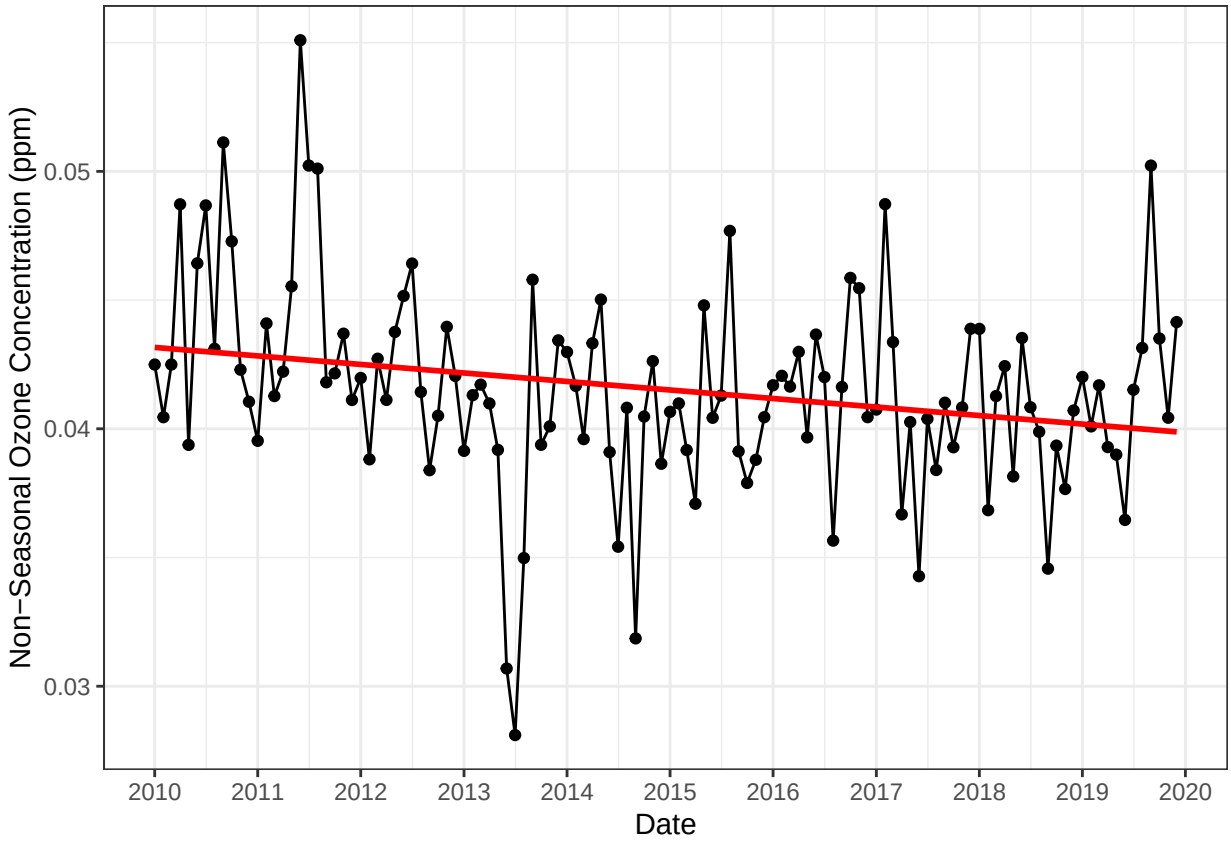
16. Run the Mann Kendall test on the non-seasonal Ozone monthly series. Compare the results with the ones obtained with the Seasonal Mann Kendall on the complete series.

```
#15
# Extract seasonal component from decomposed timeseries
GaringerOzone.monthly.trend <- as.data.frame(GaringerOzone.monthly.ts.decomposed$time.series[,1:3])

# Add observed data and date and subtract season
GaringerOzone.monthly.trend <- GaringerOzone.monthly.trend %>%
  mutate(Date = GaringerOzone.monthly$Date,
         Observed = GaringerOzone.monthly$Average_O3,
         Subtracted = Observed-GaringerOzone.monthly.trend$season)

#16
MannKendall(GaringerOzone.monthly.trend[,6])
```

```
## tau = -0.179, 2-sided pvalue =0.0037728
```

```
ggplot(GaringerOzone.monthly.trend, aes(Date, Subtracted)) +
  geom_point() +
  geom_line() +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(y = "Non-Seasonal Ozone Concentration (ppm)") +
  scale_x_date(date_breaks = "years", date_labels="%Y")
```

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



Answer: After subtracting seasonality, there is still a significant downward trend and in fact the effect is stronger (tau = -0.179, 2-sided p-value = 0.004).