Assignment 8: Time Series Analysis

Kamil Orozco

Spring 2023

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on generalized linear models.

Directions

1. Rename this file `<FirstLast>_A08_TimeSeries.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

```
#1
#getting wd and calling packages
getwd()
```

```
## [1] "/Users/kariis/Documents/EDA2"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.4     v forcats 1.0.0
## -- 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(zoo)
```

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

```
library(here)
```

```
## here() starts at /Users/kariis/Documents/EDA2
```

```
# setting ggplot theme; building then setting

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

theme_set(mytheme)
```

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
#importing datasets reading them as csv
Ozone10 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2010_raw.csv"),
  stringsAsFactors = TRUE)

Ozone11 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2011_raw.csv"),
  stringsAsFactors = TRUE)

Ozone12 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2012_raw.csv"),
  stringsAsFactors = TRUE)

Ozone13 <- read.csv(
```

```
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2013_raw.csv"),
  stringsAsFactors = TRUE)

Ozone14 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2014_raw.csv"),
  stringsAsFactors = TRUE)

Ozone15 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2015_raw.csv"),
  stringsAsFactors = TRUE)

Ozone16 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2016_raw.csv"),
  stringsAsFactors = TRUE)

Ozone17 <- read.csv(
  here("Data","Raw","Ozone_TimeSeries", "EPAair_O3_GaringerNC2017_raw.csv"),
  stringsAsFactors = TRUE)

Ozone18 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2018_raw.csv"),
  stringsAsFactors = TRUE)

Ozone19 <- read.csv(
  here("Data","Raw", "Ozone_TimeSeries", "EPAair_O3_GaringerNC2019_raw.csv"),
  stringsAsFactors = TRUE)



#Joining the datasets
OzoneList <- list(Ozone10, Ozone11, Ozone12, Ozone13, Ozone14, Ozone15, Ozone16, Ozone17, Ozone18, Ozon

GaringerOzone <- OzoneList %>% reduce(full_join)
```

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 setting date column as date class
class(GaringerOzone$Date)
```

```
## [1] "factor"
```

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


#4 wrangling columns to only have Date, Daily max 8 hour, and Daily AQI value.
ThreeVarOzone <- select(GaringerOzone, Date, DAILY_AQI_VALUE, Daily.Max.8.hour.Ozone.Concentration)

#5. creating a new data frame
Days <- as.data.frame(x= seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "1 day"))
colnames(Days) <- "Date"

#6 using left_join function to combine the data frames
GaringerOzone1 <- left_join(Days, ThreeVarOzone)
```

```
## Joining with `by = join_by(Date)`
```

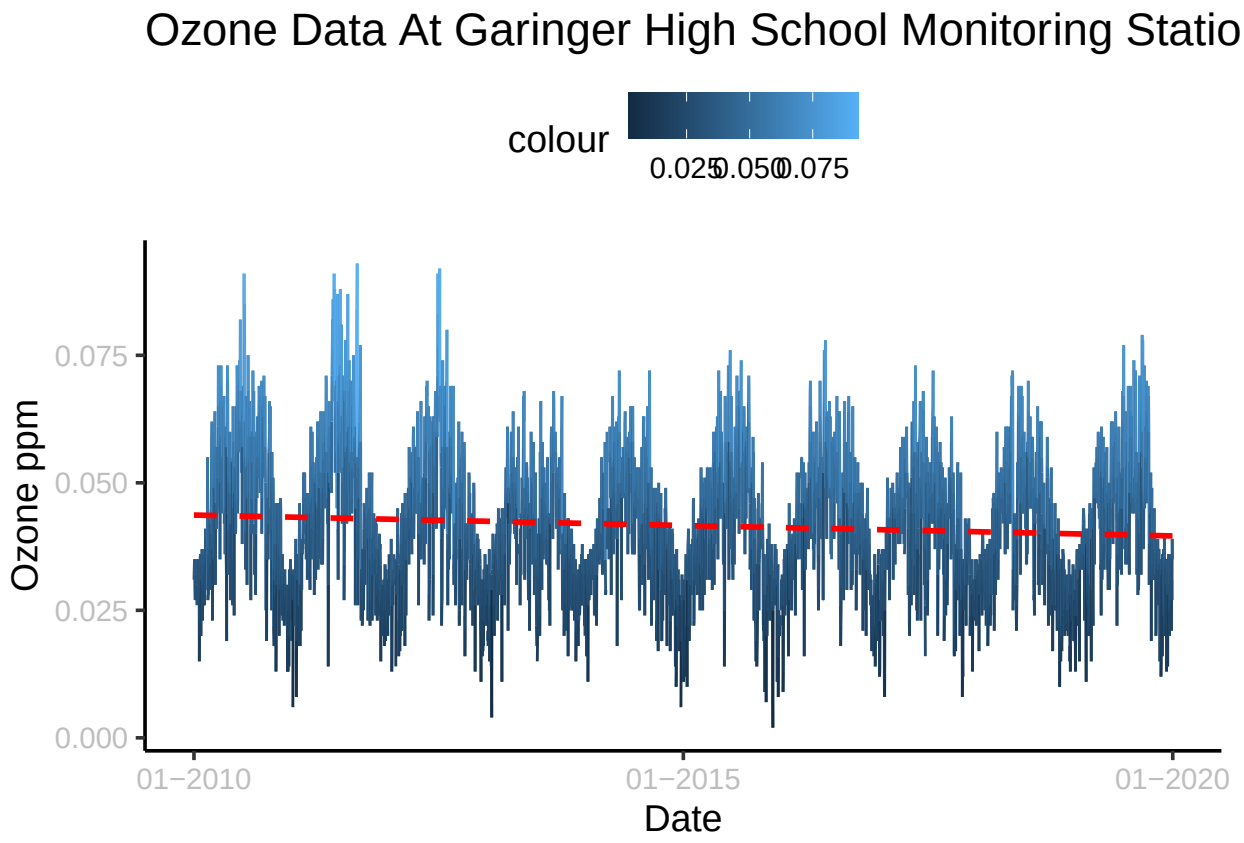
Visualize

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?

```
#7
ozonepolot <-
  ggplot(GaringerOzone1, aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration, color= "black")) +
  geom_line(aes(color=Daily.Max.8.hour.Ozone.Concentration)) +
  geom_smooth(method = lm, se= FALSE, color = 'red', linetype='dashed') +
  scale_x_date(date_labels = "%m-%Y") +
  ylab("Ozone ppm") +
  ggtitle("Ozone Data At Garinger High School Monitoring Station")
print(ozonepolot)
```

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

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



Answer: the trend is that ozone concentrations are steadily decreasing at a slow but constant rate.

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
#using linear interpolation to fill in missing data for ozone

GHSOzone <- GaringerOzone1 %>%
  mutate(OzoneConcentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: for the plot we want to produce, linear interpolation is best because 1. its a smoother degree of freedom and 2. there is less likelihood of overshooting values.

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
# creating a new data frame that contains mean ozone concentration for ea month. add columns for year a
GaringerOzone.monthly <- GHSOzone %>%
  group_by(Date, OzoneConcentration) %>%
  summarise(meanIndex = mean(OzoneConcentration))
```

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

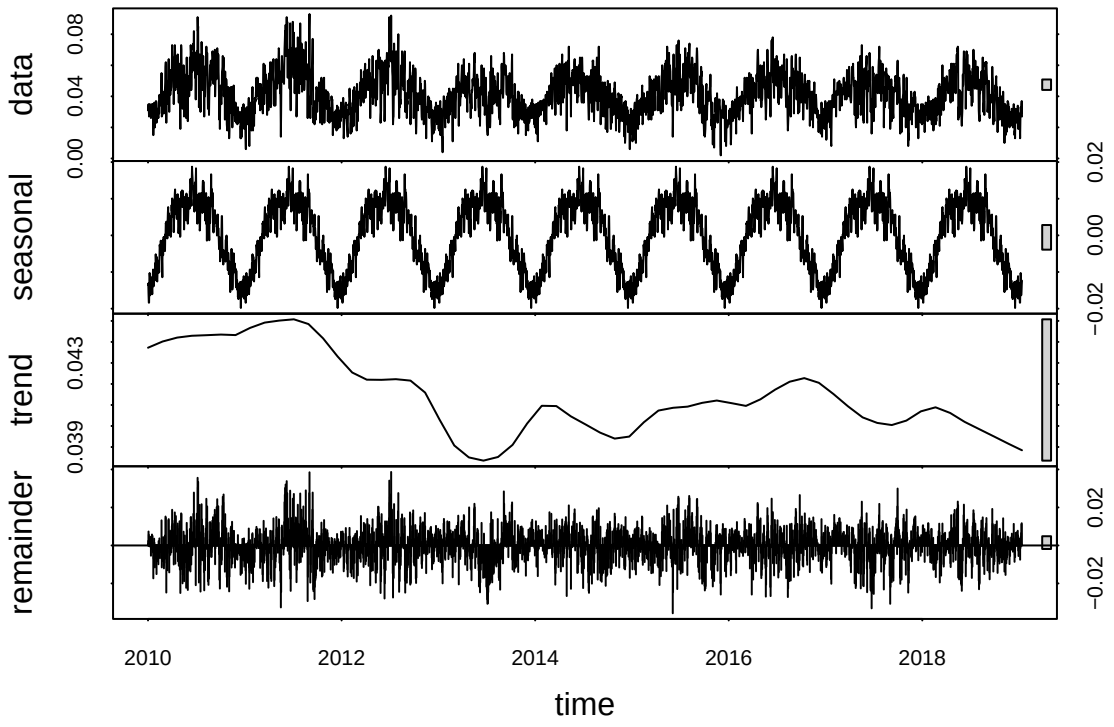
```
GaringerOzone.monthly <- mutate(GaringerOzone.monthly, month=
                                  month(Date),
                                year= year(Date))

# creating a new date column with ea month-year combo set as the first day of every month
GaringerOzone.monthly$Date <- as.yearmon(paste(GaringerOzone.monthly$year, GaringerOzone.monthly$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.

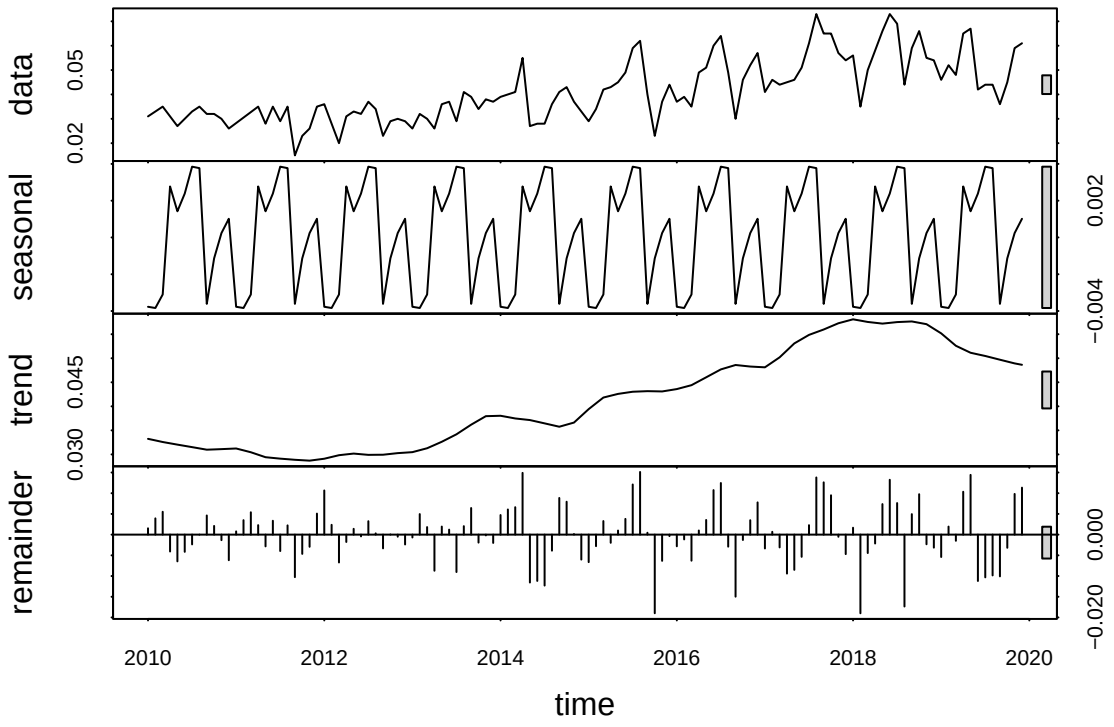
```
#10
#generating time series objects using ts function
GaringerOzone.daily.ts <- ts(GHSOzone$OzoneConcentration, frequency = 365, start = c(2010, 1), end = c(
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$meanIndex, frequency = 12, start = c(2010, 1), end
```

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

```
#11
#decomposing time series objects with stl function
DailyDecomp <- stl(GaringerOzone.daily.ts,
                   s.window = "periodic")
plot(DailyDecomp)
```




```
MonthlyDecomp <- stl(GaringerOzone.monthly.ts,
                     s.window = "periodic")
plot(MonthlyDecomp)
```



12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

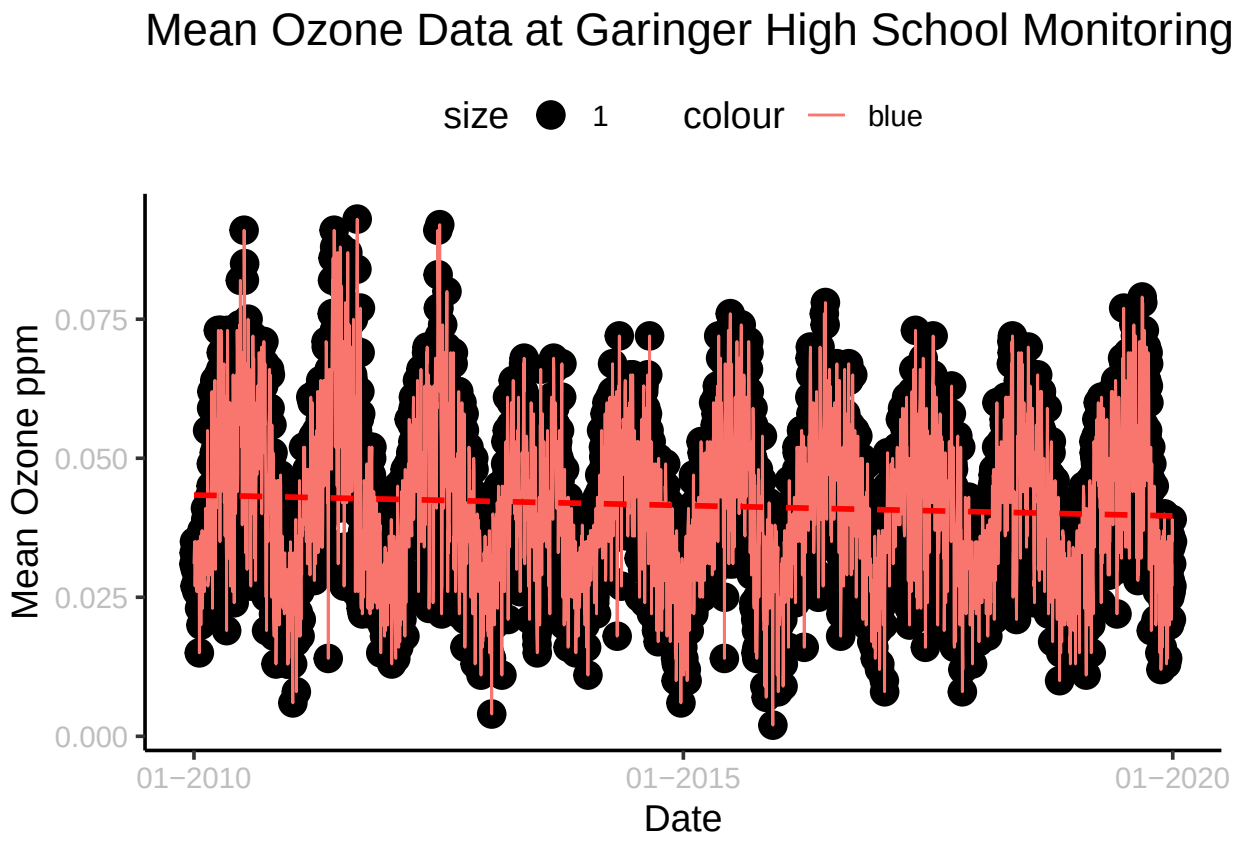
```
#12
#trend::mk.test(MonthlyDecomp, alternative = c("two.sided", "greater", "less"), continuity = TRUE)
```

Answer: The MK test doesn't require previous assumptions of the data in order to run. For example, linear regression analysis requires that there is a normal distribution along the regression line. It is best for this analysis because we are assessing the upward and downward trends of the variable of interest over time.

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
MeanOzonePlot <-
  ggplot(GHSOzone, aes(x=Date, y=OzoneConcentration)) +
  geom_point(aes(size=1)) +
  geom_line(aes(color='blue')) +
  geom_smooth(method = lm, se=FALSE, color= 'red', linetype='dashed') +
  scale_x_date(date_labels = "%m-%Y") +
  ylab("Mean Ozone ppm") +
  ggtitle("Mean Ozone Data at Garinger High School Monitoring Station")
print(MeanOzonePlot)
```

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



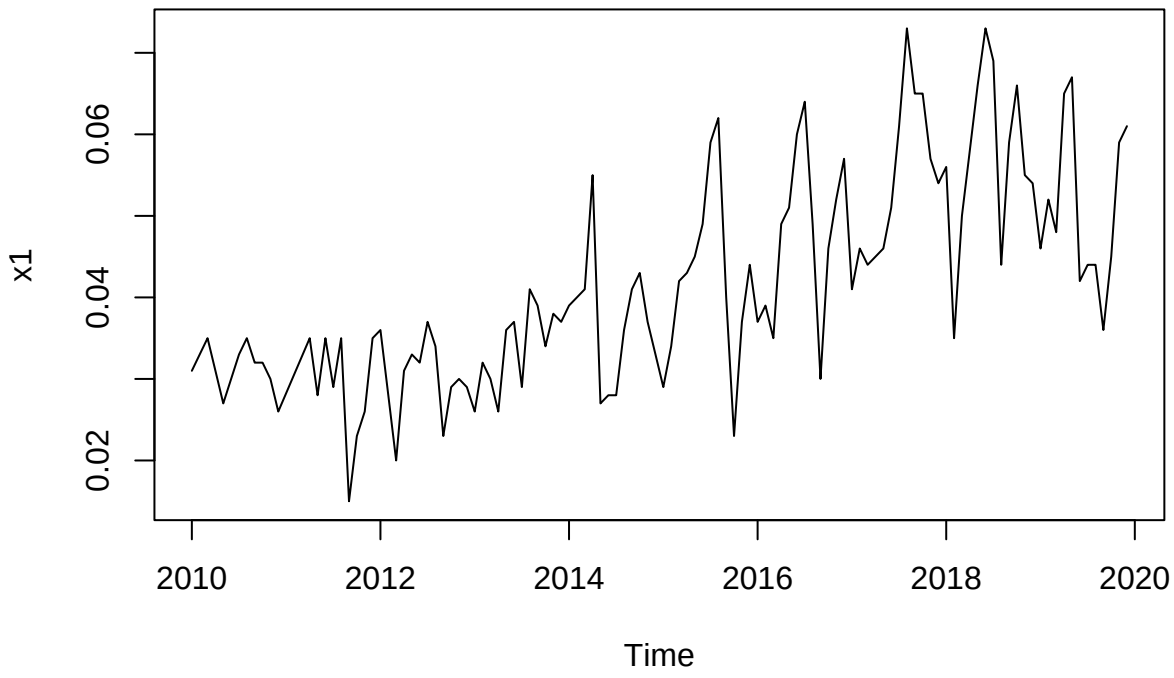
14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

    Answer: Based on these results, we can fail to reject the hypothesis meaning, the plot shows seasonally ajdusted time series (we see an average of Ozone decreasing by 1.4 ppm).

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
OzoneConc_Comp <- as.data.frame(GaringerOzone.monthly.ts, s.window = "periodic")
plot(OzoneConc_Comp)
```




```
OzoneComponents <- as.data.frame(OzoneConc_Comp$time.series[,120:1])


#16
```

Answer: