Assignment 8: Time Series Analysis

Samantha Jensen

Fall 2024

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

```
# Loading Packages
library(Kendall)
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(lubridate)
library(zoo)
```

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

```r
library(trend)
# Checking wd
here()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```r
# Setting Default Theme
theme1 <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "darkblue"),
        line = element_line(color = "black",
                            linewidth = 2),
        legend.position = "right")
theme_set(theme1)
```

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.

```r
#1
# Reading in Datasets
Ozone_10 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
           stringsAsFactors = TRUE)
Ozone_11 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
           stringsAsFactors = TRUE)
Ozone_12 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
           stringsAsFactors = TRUE)
Ozone_13 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
           stringsAsFactors = TRUE)
Ozone_14 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
           stringsAsFactors = TRUE)
Ozone_15 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
           stringsAsFactors = TRUE)
Ozone_16 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
           stringsAsFactors = TRUE)
Ozone_17 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
           stringsAsFactors = TRUE)
Ozone_18 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
           stringsAsFactors = TRUE)
Ozone_19 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
           stringsAsFactors = TRUE)

# combining Datasets
Ozone_combined <- rbind(Ozone_10, Ozone_11, Ozone_12, Ozone_13, Ozone_14,
                        Ozone_15, Ozone_16, Ozone_17, Ozone_18, Ozone_19)
```

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
# setting Date Column to date class
Ozone_combined$Date <- mdy(Ozone_combined$Date)

# 4
# selecting certain columns
Ozone_processed <- Ozone_combined %>%
      select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5
# creating a dataframe of dates from 2010 to 2019 by day
start_date <- as.Date("2010-01-01")
end_date <- as.Date("2019-12-31")
day_sequence <- seq(from = start_date, to = end_date, by = "day")
day_df <- as.data.frame(day_sequence)
# renaming column to "Date"
day_df <- day_df %>% rename(Date = day_sequence)
# 6
# Left join by date
GaringerOzone <- left_join(day_df, Ozone_processed, by = "Date")
```

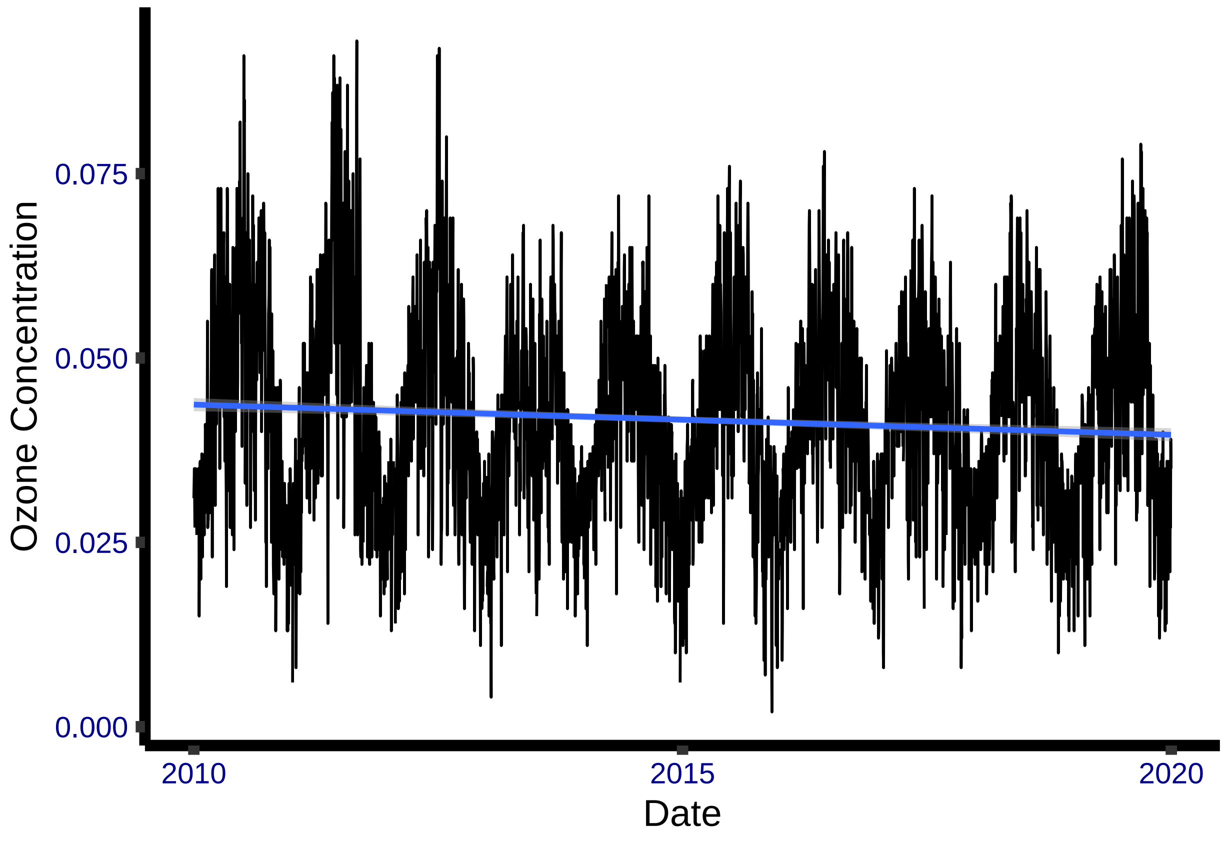
Visualize

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?

```r
#7
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = "lm") +
  ylab(expression("Ozone Concentration"))
```

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

```
## Warning: Removed 63 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```



Answer: The linear trend line shows a slight decrease in ozone concentrations 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?

```
#8
# filling in missing Ozone concentration values
GaringerOzone <- GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration = na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: Because of the strong seasonal trend in our data, it is fair to assume that the missing data points fall in between the values on either side of them, making linear interpolation a reliable method. Spline doesn't make sense because a quadratic function isn't necessary for this dataset. Piecewise uses the "nearest neighbor" which doesn't work because there is data from everyday, so there isn't a nearest neighbor because the days are evenly spaced.

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
# New df with columns for month and year
GaringerOzone.monthly <- GaringerOzone %>%
    mutate(Month = month(Date), Year = year(Date))
# taking the mean from each month of each year
GaringerOzone.monthly <- GaringerOzone.monthly %>%
   group_by(Month, Year) %>%
  summarise(Daily.Max.8.hour.Ozone.Concentration = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  rename(Monthly_average = Daily.Max.8.hour.Ozone.Concentration)
```

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

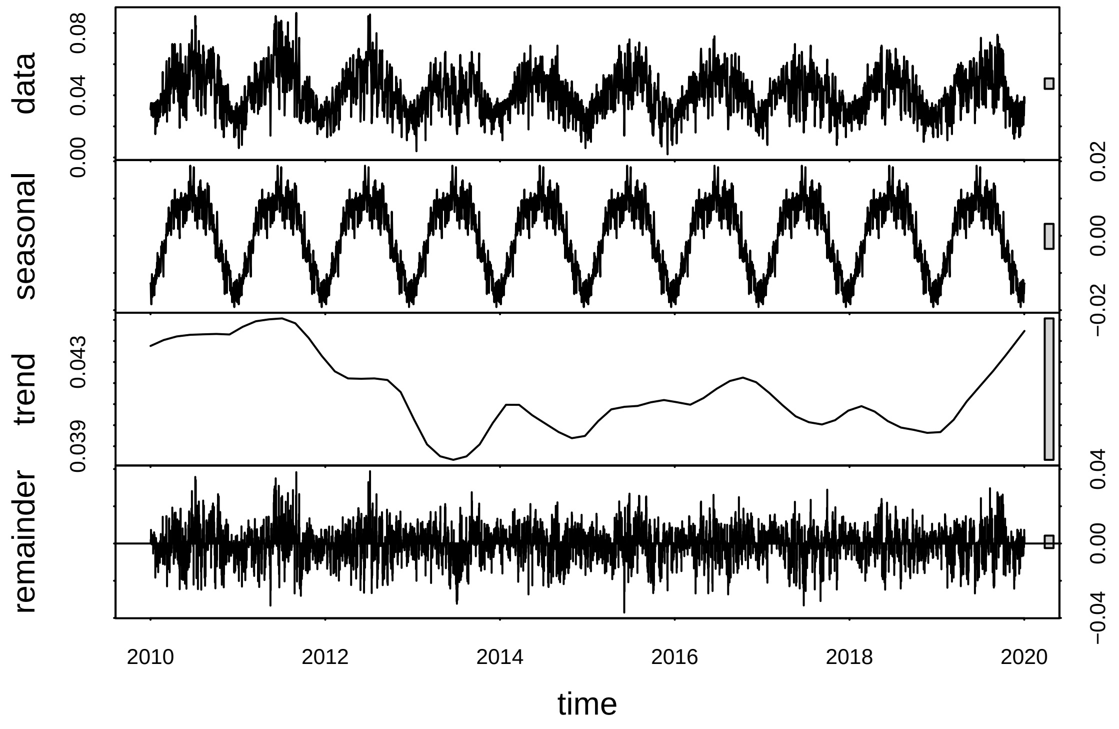
```
  # new column for plotting with the first day of each month per year as the label
GaringerOzone.monthly <- mutate(GaringerOzone.monthly,
                               Date = as.Date(
                                paste(Year, Month, "01", sep = "-")))
```

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.

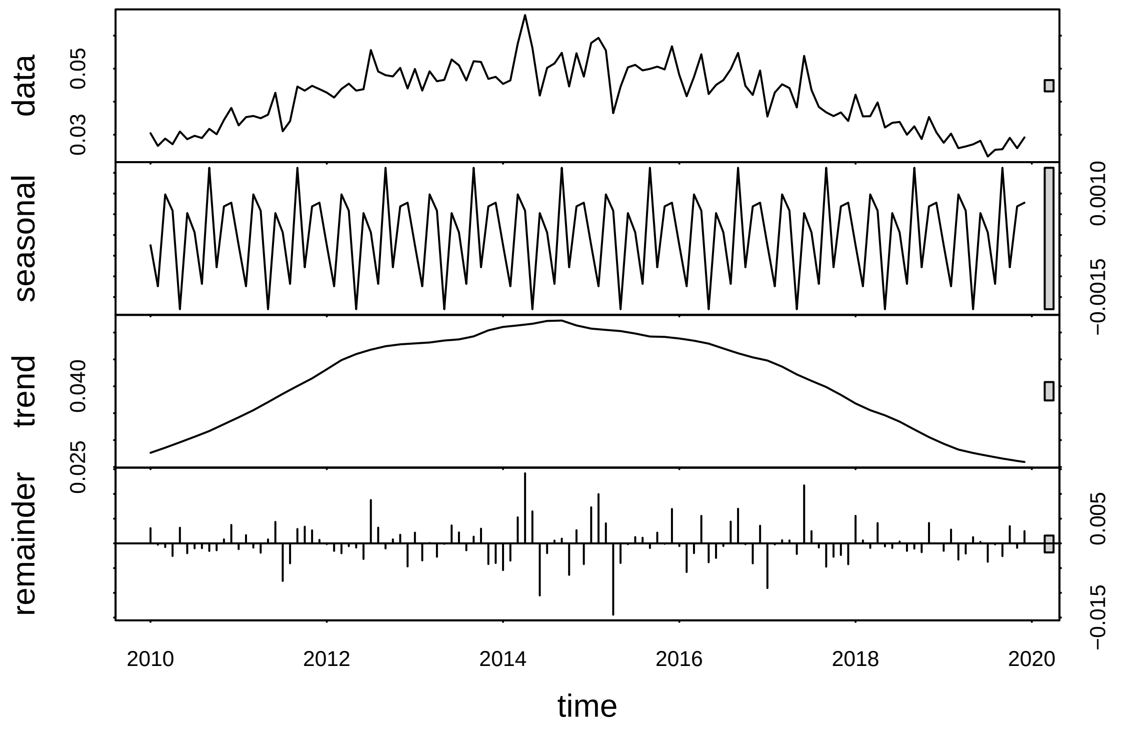
```
#10
# Time series for daily observations
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010, 01), frequency = 365)
# Time series for Monthly average concentrations
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Monthly_average,
                               start = c(2010, 01), frequency = 12)
```

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

```r
#11
# decomposing daily and monthly time series objects
GaringerDaily_decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerDaily_decomposed)
```



```r
GaringerMonthly_decomposed <- stl(GaringerOzone.monthly.ts,
                                  s.window = "periodic")
plot(GaringerMonthly_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?

```
#12
# running the Seasonal Mann Kendall trend Analysis
SeasonalMannKendall(GaringerOzone.monthly.ts)
```

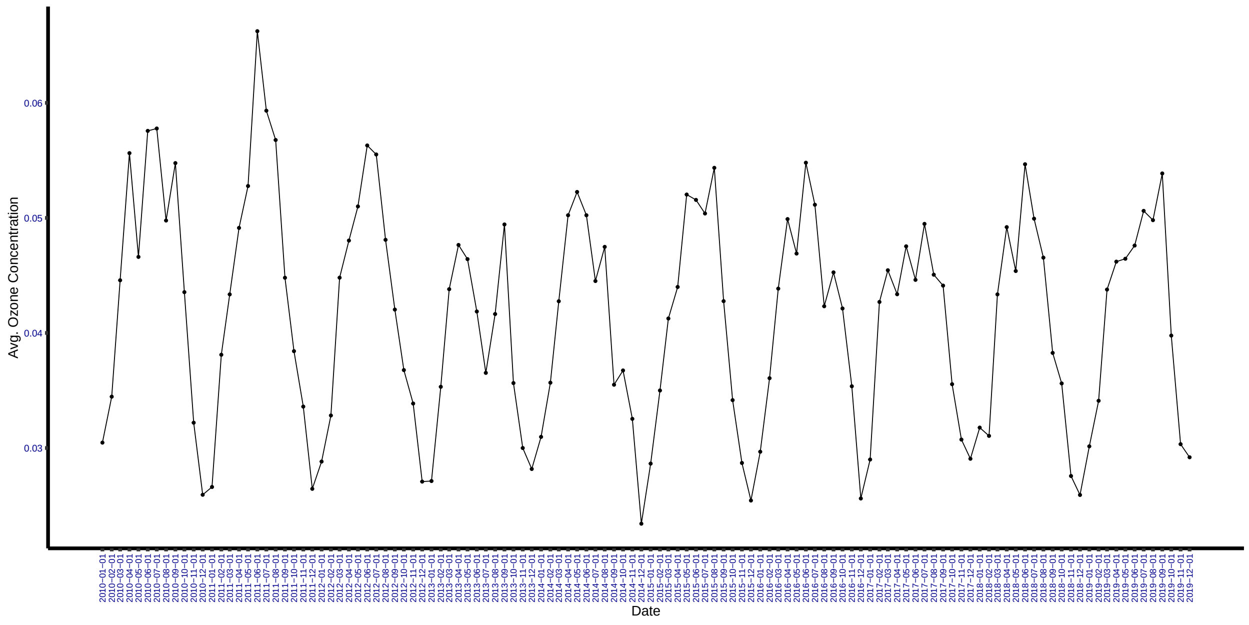
```
## tau = -0.1, 2-sided pvalue =0.16323
```

Answer: The seasonal Mann-Kendall is the most approporiate because it is the only monotonic trend analysis that can handle seasonal trends.

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
# plotting monthly ozone concentration over time
ggplot(GaringerOzone.monthly, aes(x = Date, y = Monthly_average)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks = unique(GaringerOzone.monthly$Date),
                     labels = format(unique(GaringerOzone.monthly$Date),
                                     nsmall = .5))  +
  labs(x = "Date", y = "Avg. Ozone Concentration") +
```

```r
  theme(axis.text.x = element_text(angle= 90, vjust = .5, hjust = 1, size = 10),
        axis.title.x = element_text(size = 16),
        axis.title.y = element_text(size = 16))
```



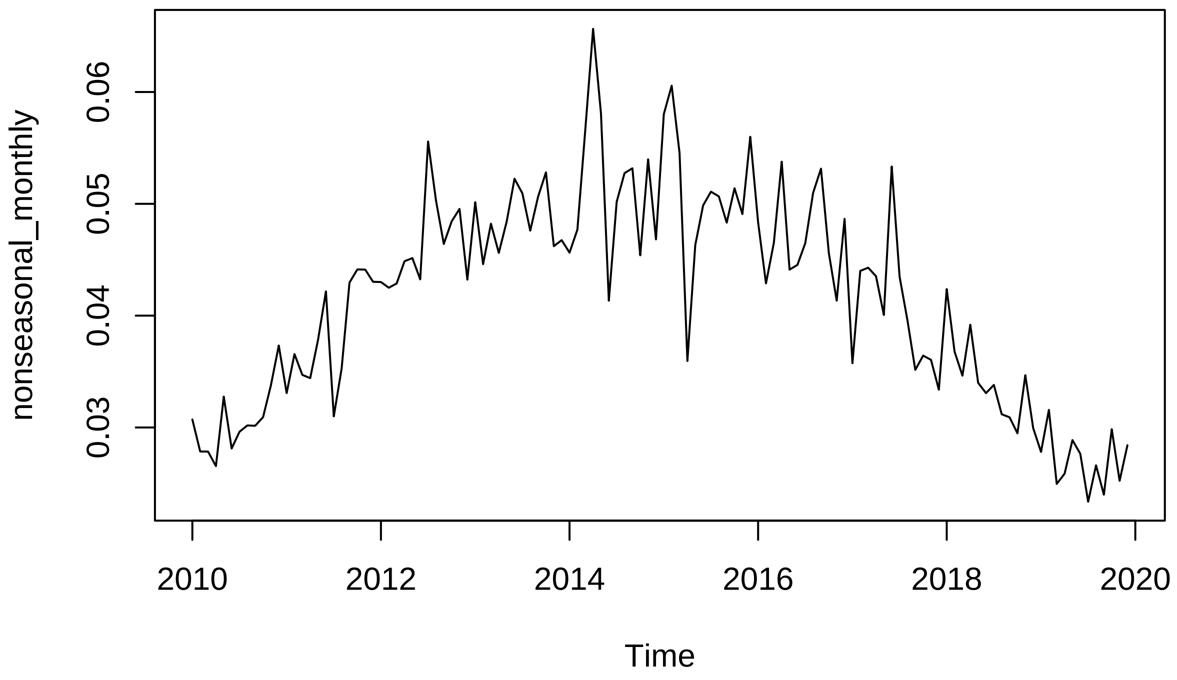
14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

    Answer: This graph shows that there has been a very small deacrease in ozone concentrations based on a linear model from 2010 to 2020. The results from the seasonal Mann Kendall test support this small negative strend, with a tau value of -0.1. Since the tau value is so close to 0, this trend is very weak. The p-value of 0.16232 is greater than 0.05 indicates that this trend is not statistically significant. (tau = -0.1, 2-sided pvalue =0.16323)

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.

```r
#15
# Removing the seasonal component from the Monthly time series
nonseasonal_monthly <- GaringerOzone.monthly.ts -
  GaringerMonthly_decomposed$time.series[,"seasonal"]
plot(nonseasonal_monthly)
```



```
#16
#running Mann Kendall Test on the time series without seasonal component
MannKendall(nonseasonal_monthly)
```

```
## tau = -0.101, 2-sided pvalue =0.10388
```

Answer: The tau value of both tests is roughly -0.1 indicating a weak or no trend. The pvalue of the Mann Kendall analysis is smaller than the seasonal Pvalue, but is still not statistically significant to support that there has been a significant change in ozone concentration from 2010 to 2020.