Assignment 7: Time Series Analysis

Clara Fast

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()

#Load required packages
require(tidyverse)
```

```
## Loading required package: 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   1.1.3     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()
```

```
require(lubridate)
```

```
## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
require(zoo)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

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

```
require(trend)
```

```
## Loading required package: trend
```

```
require(dplyr)

#Set theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        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
Ozone_2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                       stringsAsFactors = TRUE)
Ozone_2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                       stringsAsFactors = TRUE)
Ozone_2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                       stringsAsFactors = TRUE)
Ozone_2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                       stringsAsFactors = TRUE)
Ozone_2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
```

```
                      stringsAsFactors = TRUE)
Ozone_2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                      stringsAsFactors = TRUE)
Ozone_2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                      stringsAsFactors = TRUE)
Ozone_2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                      stringsAsFactors = TRUE)
Ozone_2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                      stringsAsFactors = TRUE)
Ozone_2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                      stringsAsFactors = TRUE)


#Combine into single Dataframe
GaringerOzone <- rbind(Ozone_2010, Ozone_2011, Ozone_2012, Ozone_2013,
                           Ozone_2014, Ozone_2015, Ozone_2016, Ozone_2017,
                           Ozone_2018, Ozone_2019)
```

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
#Check class of Date column
class(GaringerOzone$Date)
```

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

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

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

# 5
#Generate Daily Dataframe
Daily <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"),
                           by = "day"))
```

```
#Rename Date column
names(Daily)[1] <- 'Date'

# 6
#Combine Dataframes
GaringerOzone<-left_join(Daily, GaringerOzone_processed)
```

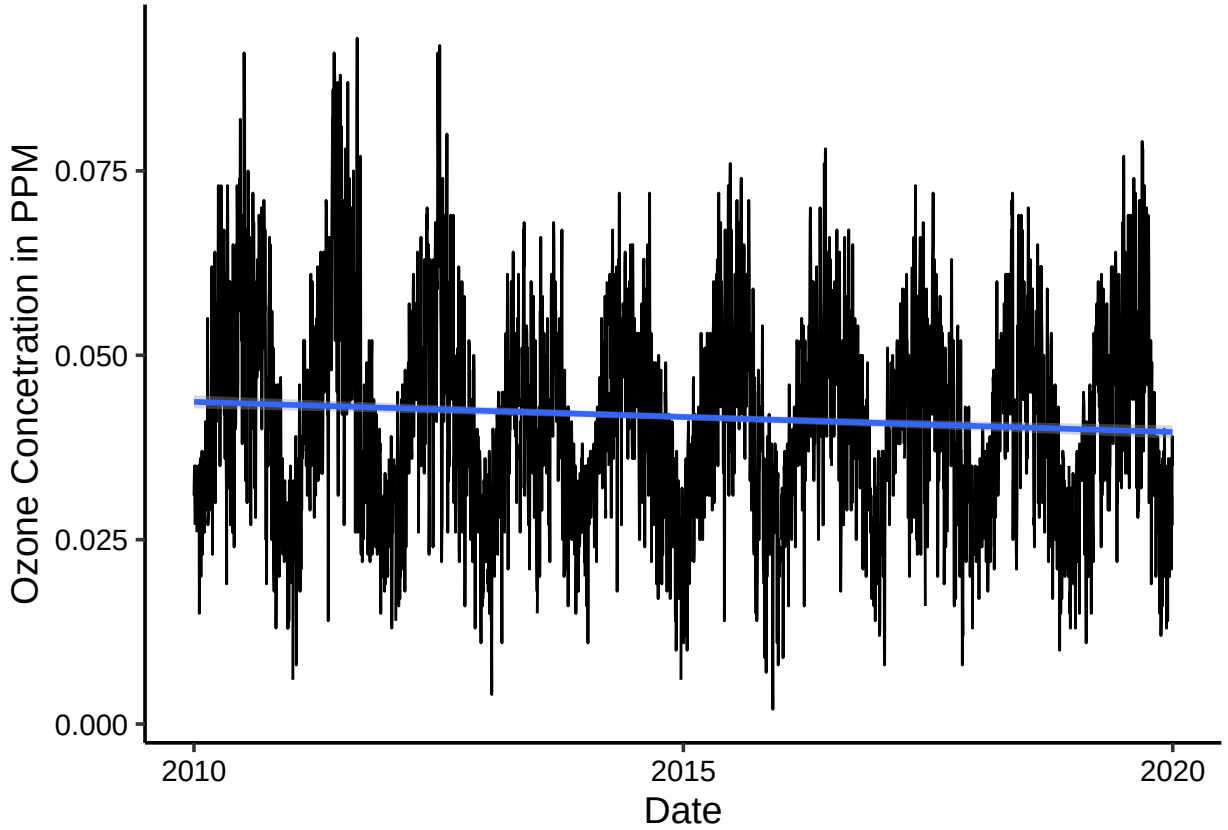
```
## Joining, by = "Date"
```

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?

```
#7
#Create line plot depicting ozone concentration over time
lineplot_ozone <- ggplot(GaringerOzone,
                         aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = "lm", formula= y~x) + ylab("Ozone Concetration in PPM") +
  xlab("Date")

print(lineplot_ozone)
```

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



Answer: The plot suggests a slightly downward trend in ozone concentration over time, as indicated by the trend line.

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
#Fill in missing daily data for ozone concentration
GaringerOzone_8 <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration=zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: We did not use a piecewise constant or spline interpolation because as we saw from the previous analysis, there is a linear downward trend. Piecewise constant and spline interpolation both assume that any missing data are equal to the measurement nearest to it, which would cause a plateau in the trend. Further, as the distance between the missing data and values that are not missing will always be one day, it makes most sense to use linear interpolation, as it would connect the dots between the day before and the day after, in essence finding the average between the two. This approach makes most sense concerning the structure of the data, and the pre-existing trend.

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
#Create new Dataframe with aggregated data
GaringerOzone.monthly<-
  GaringerOzone_8 %>%
  mutate(Month = month(Date),
         Year = year(Date)) %>%
  mutate(Month_Year = my(paste0(Month, "-", Year))) %>%
  dplyr::group_by(Month,Year, Month_Year) %>%
  dplyr:: summarise(MeanOzone = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

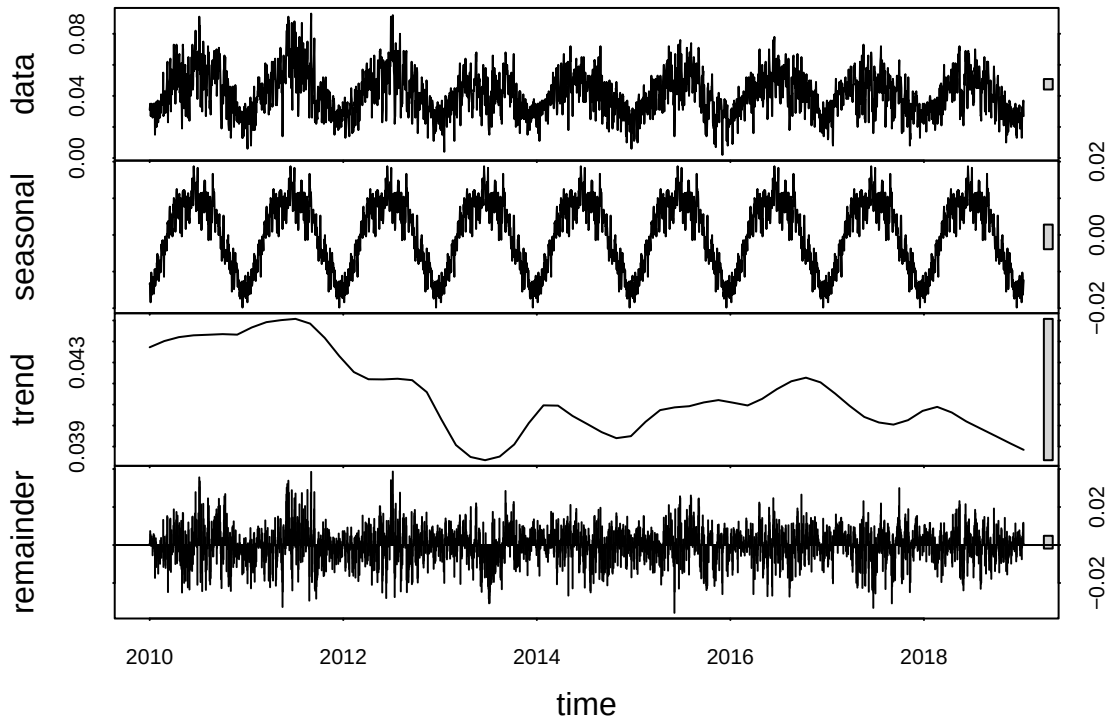
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

```
#10
#Generate time series for daily ozone observations
GaringerOzone.daily.ts<-ts(GaringerOzone_8$Daily.Max.8.hour.Ozone.Concentration,
                           start = c(2010,01,01), end = c(2019,12,01),
```

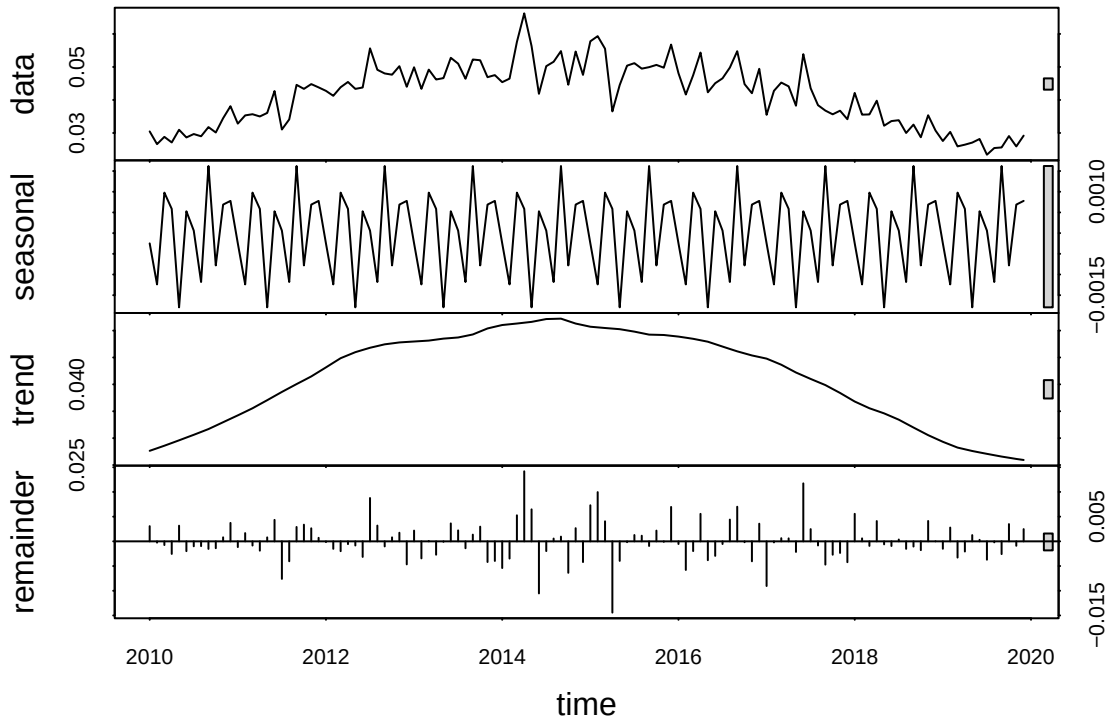
```
                              frequency = 365)


#Generate time series for monthly average ozone values
GaringerOzone.monthly.ts<-ts(GaringerOzone.monthly$MeanOzone,
                             start = c(2010,01,01), end = c(2019,12,01),
                             frequency = 12)
```

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

```
#11
#Decompose daily time series
daily_stl <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(daily_stl)
```




```
#Decompose monthly time series
monthly_stl <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(monthly_stl)
```



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
#Run Mann-Kendall test on seasonal monthly ozone series
require("Kendall")
```

```
## Loading required package: Kendall
```

```
MK_monthlyozone<-SeasonalMannKendall(GaringerOzone.monthly.ts)
print(MK_monthlyozone)
```

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

```
summary(MK_monthlyozone)
```

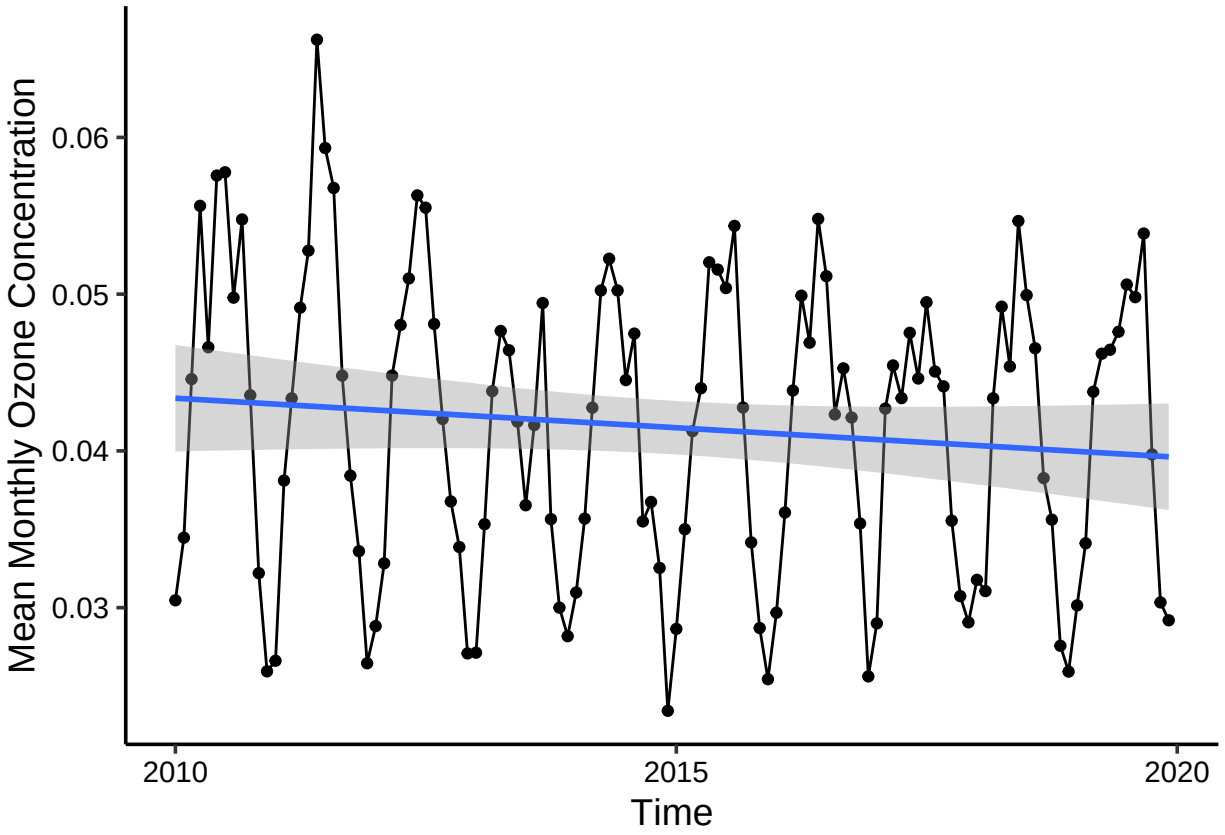
```
## Score =  -54 , Var(Score) = 1500
## denominator =  540
## tau = -0.1, 2-sided pvalue =0.16323
```

Answer: Monotonic trends are a gradual shift over time that is consistent in direction, which is fitting for the monthly Ozone series as there appears to be a consistent downward trend. The seasonal Mann-Kendall test is used to analyze data that follow a monotonic trend. It allows for missing data, non-normality, and is used for data that is collected seasonally. As we have not yet factored out the seasonality of the data, it would be appropriate to use this test.

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
#Create plot depicting mean monthly ozone concentrations over time
ozone_plot <-
ggplot(GaringerOzone.monthly, aes(x = Month_Year, y = MeanOzone)) +
  geom_point() +
  geom_line() +
  ylab("Mean Monthly Ozone Concentration") +
  xlab("Time") +
  geom_smooth(method = lm)
print(ozone_plot)
```

```
## `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: The data suggests that ozone concentrations have steadily decreased over the 2010s. This is observed by the graph above in the trend line, that follows mean monthly ozone concentration over time. The Mann-Kendall test however does not support this conclusion, demonstrating a statistically insignificant negative trend (tau = -0.1, 2-sided pvalue = 0.16323). While a negative trend is shown by the test, it is insignificant.

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
#Subtract seasonal component from monthly ozone time series
GaringerOzone.monthly.subtract <- as.data.frame(monthly_stl$time.series[,1:3])

GaringerOzone.monthly.subtract <- mutate(GaringerOzone.monthly.subtract,
        Observed = GaringerOzone.monthly$MeanOzone,
        Date = GaringerOzone.monthly$Month_Year)

#Convert to time series
Ozonemonthly_newts<-ts(GaringerOzone.monthly.subtract$Observed,
                           start = c(2010,01,01), end = c(2019,12,01),
                           frequency = 12)


#16
#Run Mann-Kendall test on non-seasonal ozone monthly series
MK_monthlyozone_new<-Kendall::MannKendall(Ozonemonthly_newts)
print(MK_monthlyozone_new)
```

```
## tau = -0.105, 2-sided pvalue =0.088483
```

```
summary(MK_monthlyozone_new)
```

```
## Score =  -752 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.105, 2-sided pvalue =0.088483
```

Answer: There is no stark difference in the conclusions extrapolated from this new Mann Kendall test that was generated for the non-seasonal Ozone monthly series. The test demonstrates a negative trend, but it is still statistically insignificant (tau = -0.105, 2-sided pvalue =0.088483). While it is approaching more statistically significant than the previous test, it is still above our alpha of 0.05.