Assignment 8: Time Series Analysis

Quinn Bankson

Fall 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

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.2.1     v dplyr   1.1.2
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.4     v forcats 0.5.2
## -- 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(trend)
library(Kendall)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(here)
```

```
## here() starts at /Users/mac/Documents/EDE_Fall2023
```

```
library(purrr)
library(dplyr)
here
```

```
## function (...)
## {
##     .root_env$root$f(...)
## }
## <bytecode: 0x7fd4ac383080>
## <environment: namespace:here>
```

```
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.

```
#1
file_names <- c(
  "/Users/mac/Documents/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
  "/Users/mac/Documents/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
  "/Users/mac/Documents/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
  "/Users/mac/Documents/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
  "/Users/mac/Documents/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
  "/Users/mac/Documents/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
```

```
  "/Users/mac/Documents/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
  "/Users/mac/Documents/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
  "/Users/mac/Documents/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
  "/Users/mac/Documents/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"
)

data_list <- lapply(file_names, read.csv, header = TRUE)
GaringerOzone <- do.call(rbind, data_list)

dim(GaringerOzone)
```

```
## [1] 3589   20
```

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$Date <- mdy(GaringerOzone$Date)


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

# 5
start = ymd("2010-01-01")
end = ymd("2019-12-31")
Days <- as.data.frame(seq(start, end, by = "days"))
colnames(Days) <- "Date"

# 6
class(GaringerOzone$Date)
```

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

```
class(Days$Date)
```

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

```
str(GaringerOzone)
```

```
## '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 ...
```

```
str(Days)
```

```
## 'data.frame':    3652 obs. of  1 variable:
##  $ Date: Date, format: "2010-01-01" "2010-01-02" ...
```

```
#left_join gave me some issues that I could not solve! Not sure why.
#All of the values in the rows beyond date turned to NA no matter what I tried
# with left_join.
GaringerOzone <- full_join(Days, GaringerOzone, by = "Date")
```

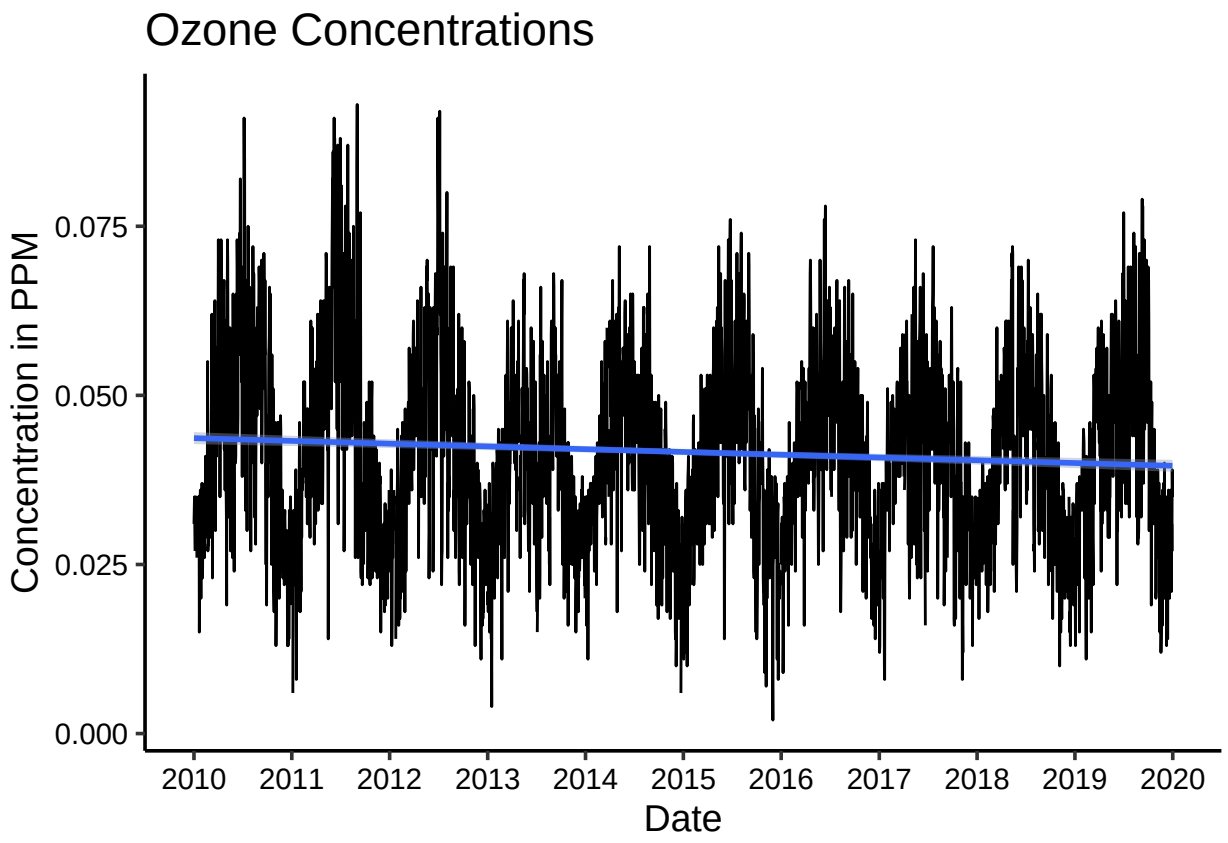
Visualize

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
ppm_concentrations <- ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = "lm") +
  labs(title = "Ozone Concentrations", x = "Date", y = "Concentration in PPM")+
  scale_x_date(date_labels = "%Y", date_breaks = "1 year")

ppm_concentrations
```

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

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



Answer: Yes. Ozone has cyclical trends every year, but ultimately the concentration is decreasing over the time periods analyzed.

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$Daily.Max.8.hour.Ozone.Concentration <- na.approx(
  GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
  x = GaringerOzone$Date
)
any(is.na(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration))
```

```
## [1] FALSE
```

Answer: There is too much variance in ozone data for us to assume a piecewise constant could work. There is not evidence that data for the NAs would be constant between the known data points they are between. Spline suits more complex and unpredictable data. Ozone data is following a trend that we can see in the time series and we should feel reasonably confident in using a relatively simple linear interpolation to fill in the missing data.

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) %>%
  summarise(mean_concentration = mean(Daily.Max.8.hour.Ozone.Concentration, na.rm = TRUE)) %>%
  ungroup() %>%
  mutate(Date.FD = ymd(paste(year, month, "01")))
```

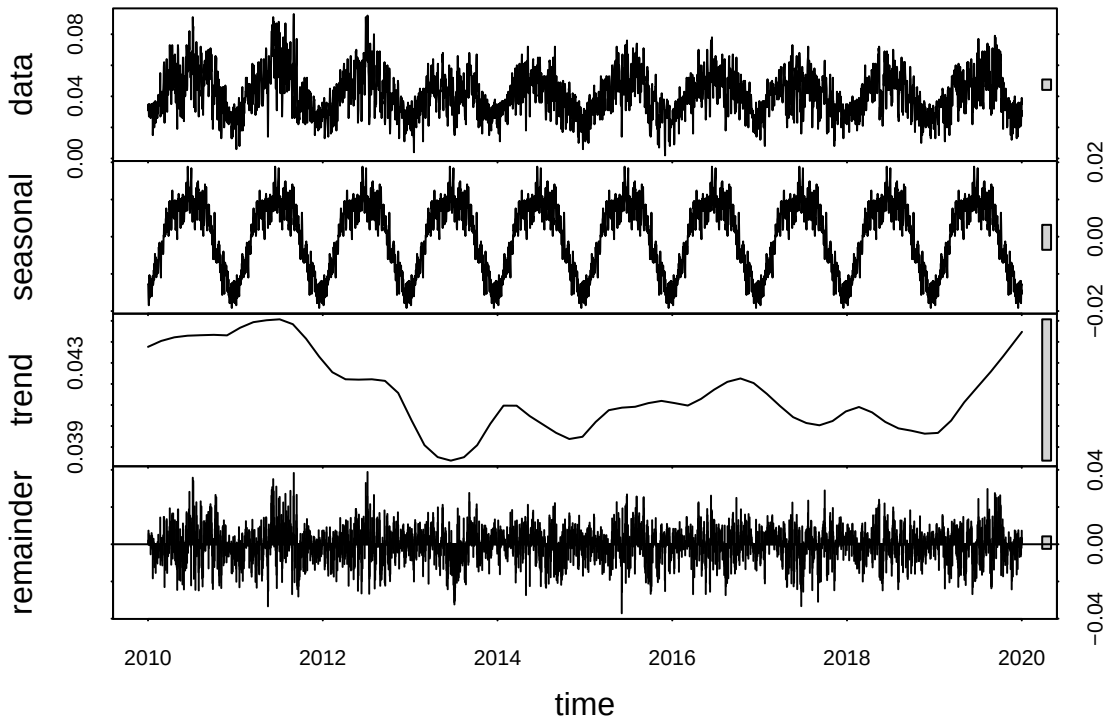
```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

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.

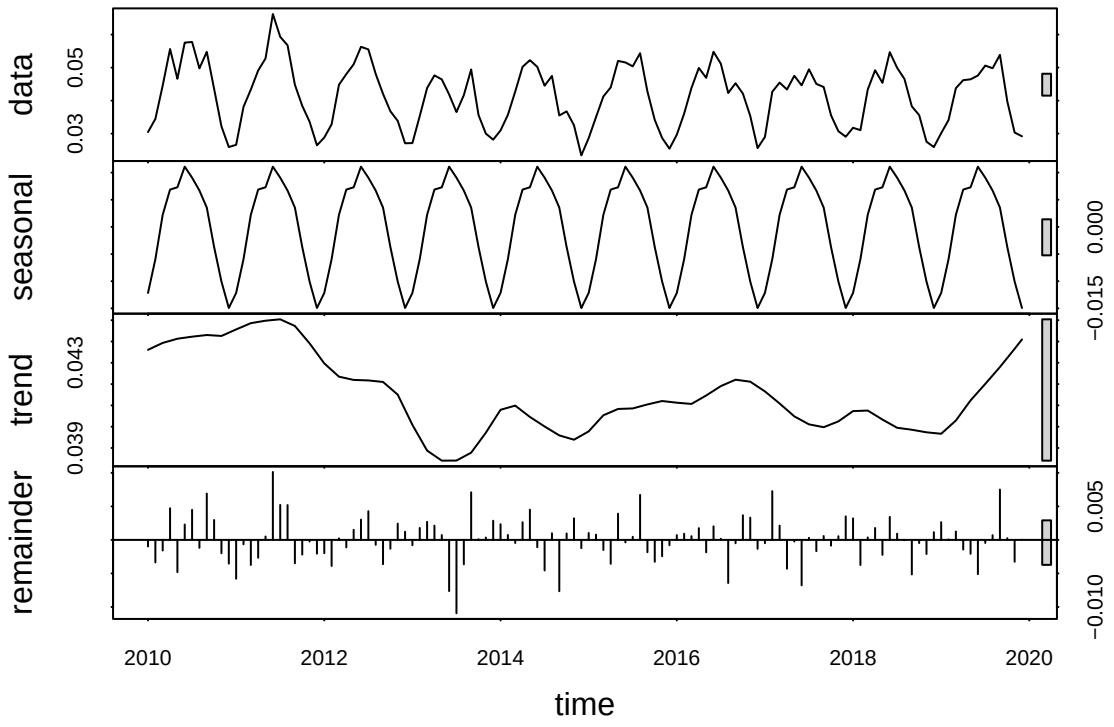
```
#10
fday.d <- day(first(GaringerOzone$Date))
fyear.d<- year(first(GaringerOzone$Date))

fmonth.m <- month(first(GaringerOzone.monthly$Date.FD))
fyear.m<- year(first(GaringerOzone.monthly$Date.FD))

GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration, start =c(fyear.d, fday
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_concentration, start = c(fyear.m, fmonth.m),
```

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

```
#11
GaringerOzone.DailyDecomposed <- stl(GaringerOzone.daily.ts, s.window = "per")
plot(GaringerOzone.DailyDecomposed)
```




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



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
mono.trend.results <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
print(mono.trend.results)
```

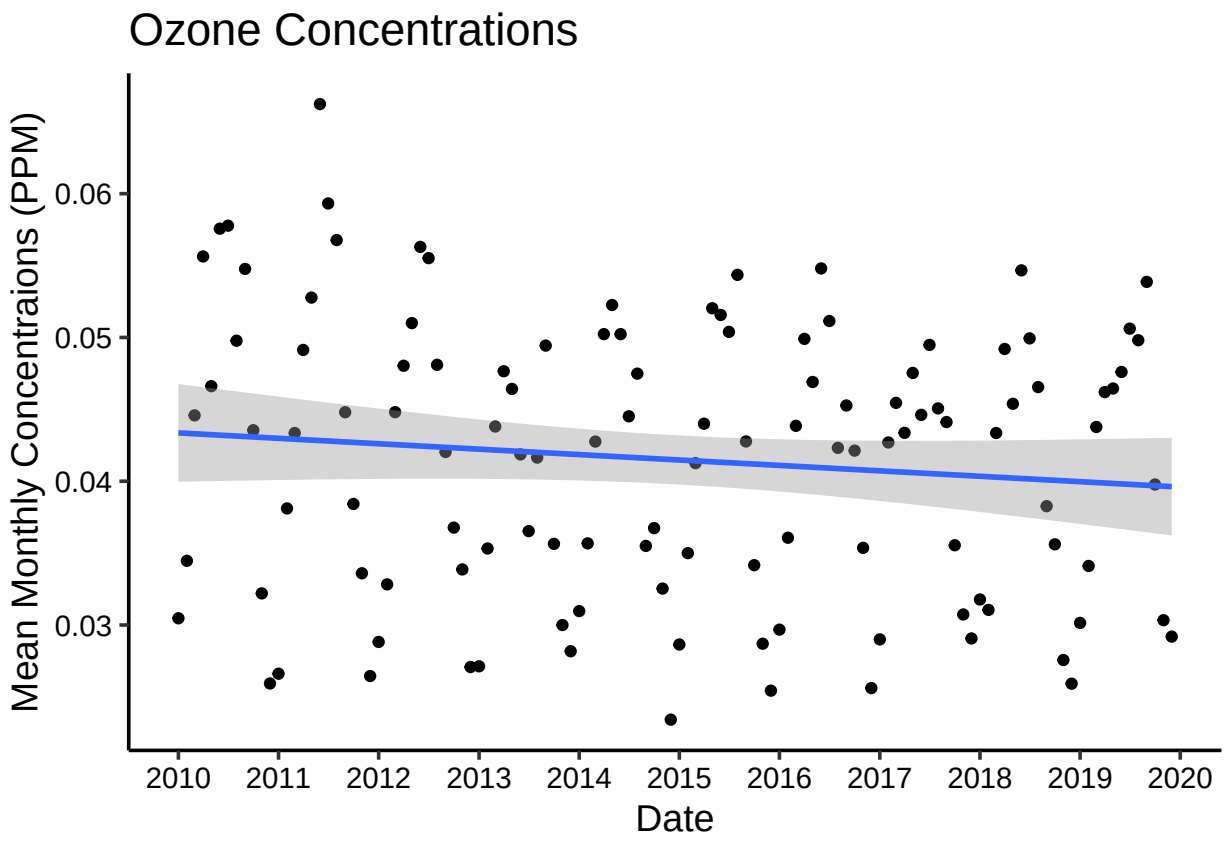
```
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The ozone quanities are following a gradual downward trend but they fluctuate frequently. SMK is appropriate to capture the long term "linear pattern" while delaing with seasonal/cyclical influences on the patters in the data.

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
ppm_mean_monthly <- ggplot(GaringerOzone.monthly, aes(x = Date.FD, y = mean_concentration)) +
  geom_point() +
  geom_smooth(method = "lm", ) +
  labs(title = "Ozone Concentrations", x = "Date", y = "Mean Monthly Concentraions (PPM)")+
  scale_x_date(date_labels = "%Y", date_breaks = "1 year")
ppm_mean_monthly
```

```
## `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: There is a slight decreasing trend in mean monthly ozone concentrations overtime. This decreasing trend is statistically significant according to a seasonal Mann-Kendall test. Despite the seasonsal influctiations in the data, an overall decreasing trend is supported by a tau of -.143 with a p value of 0.0467. (tau = -0.143, 2-sided pvalue =0.046724)

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
GaringerOzone.components <- as.data.frame(GaringerOzone.MonthlyDecomposed$time.series[,1:3])

GaringerOzone.components <- mutate(GaringerOzone.components,
      Observed = GaringerOzone.monthly$mean_concentration,
      Date = GaringerOzone.monthly$Date.FD,
      Non.Szn = GaringerOzone.components$trend + GaringerOzone.components$remainder)
```

```r
GaringerOzone.NonSzn.ts <-  ts(GaringerOzone.components$Non.Szn, start = c(fyear.m, fmonth.m), frequenc

#16

mono.non.szn.results <- Kendall::MannKendall(GaringerOzone.NonSzn.ts)
print(mono.non.szn.results)
```

```
## tau = -0.165, 2-sided pvalue =0.0075402
```

Answer: The results are not different if you use a seasonal Mann-Kendall on the non seasonal data. However, if you use a non seasonal Mann-Kandall on the non seasonal components of the montly data, we can see that tau = -0.165 and the 2-sided pvalue =0.0075402. This has a much higher statistical significance than the last tau, but a similar magnitude.