Assignment 8: Time Series Analysis

Tristen Townsend

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics (ENV872L) on time series analysis.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Use the lesson as a guide. It contains code that can be modified to complete the assignment.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document. Space for your answers is provided in this document and is indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file. You will need to have the correct software installed to do this (see Software Installation Guide) Press the `Knit` button in the RStudio scripting panel. This will save the PDF output in your Assignments folder.
6. After Knitting, please submit the completed exercise (PDF file) to the dropbox in Sakai. Please add your last name into the file name (e.g., "Salk_A08_TimeSeries.pdf") prior to submission.

The completed exercise is due on Tuesday, 19 March, 2019 before class begins.

Brainstorm a project topic

1. Spend 15 minutes brainstorming ideas for a project topic, and look for a dataset if you are choosing your own rather than using a class dataset. Remember your topic choices are due by the end of March, and you should post your choice ASAP to the forum on Sakai.

Question: Did you do this?

> ANSWER: Yes, but I haven't decided my topic quite yet.

Set up your session

2. Set up your session. Upload the EPA air quality raw dataset for PM2.5 in 2018, and the processed NTL-LTER dataset for nutrients in Peter and Paul lakes. Build a ggplot theme and set it as your default theme. Make sure date variables are set to a date format.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0
```

```
## Warning: package 'tibble' was built under R version 3.5.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(forcats)
library(lubridate)
```

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

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(pander)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(RColorBrewer)
library(colormap)
library(ggpubr)
```

```
## Loading required package: magrittr
```

```
##
## Attaching package: 'magrittr'
```

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

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(trend)
library(nlme)
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
library(lsmeans)
```

```
## Loading required package: emmeans
```

```
## The 'lsmeans' package is now basically a front end for 'emmeans'.
## Users are encouraged to switch the rest of the way.
## See help('transition') for more information, including how to
## convert old 'lsmeans' objects and scripts to work with 'emmeans'.
```

```
library(multcompView)


#1
getwd()
```

```
## [1] "/Users/Tristen/OneDrive - Duke University/Spring 2019/Data Analytics/Environmental_Data_Analyti
```

```
EPAair.2018 <- read.csv("./Data/Raw/EPAair_PM25_NC2018_raw.csv")
NTLnutrients <- read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaul_Processed.csv")

#2
tristentheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")

class(EPAair.2018$Date)
```

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

```
class(NTLnutrients$sampledate)
```

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

```
EPAair.2018$Date <- as.Date(EPAair.2018$Date,
                                                format = "%m/%d/%y")
NTLnutrients$sampledate <- as.Date(NTLnutrients$sampledate,
                                                format = "%Y-%m-%d")
```

Run a hierarchical (mixed-effects) model

Research question: Do PM2.5 concentrations have a significant trend in 2018?

3. Run a repeated measures ANOVA, with PM2.5 concentrations as the response, Date as a fixed effect, and Site.Name as a random effect. This will allow us to extrapolate PM2.5 concentrations across North Carolina.

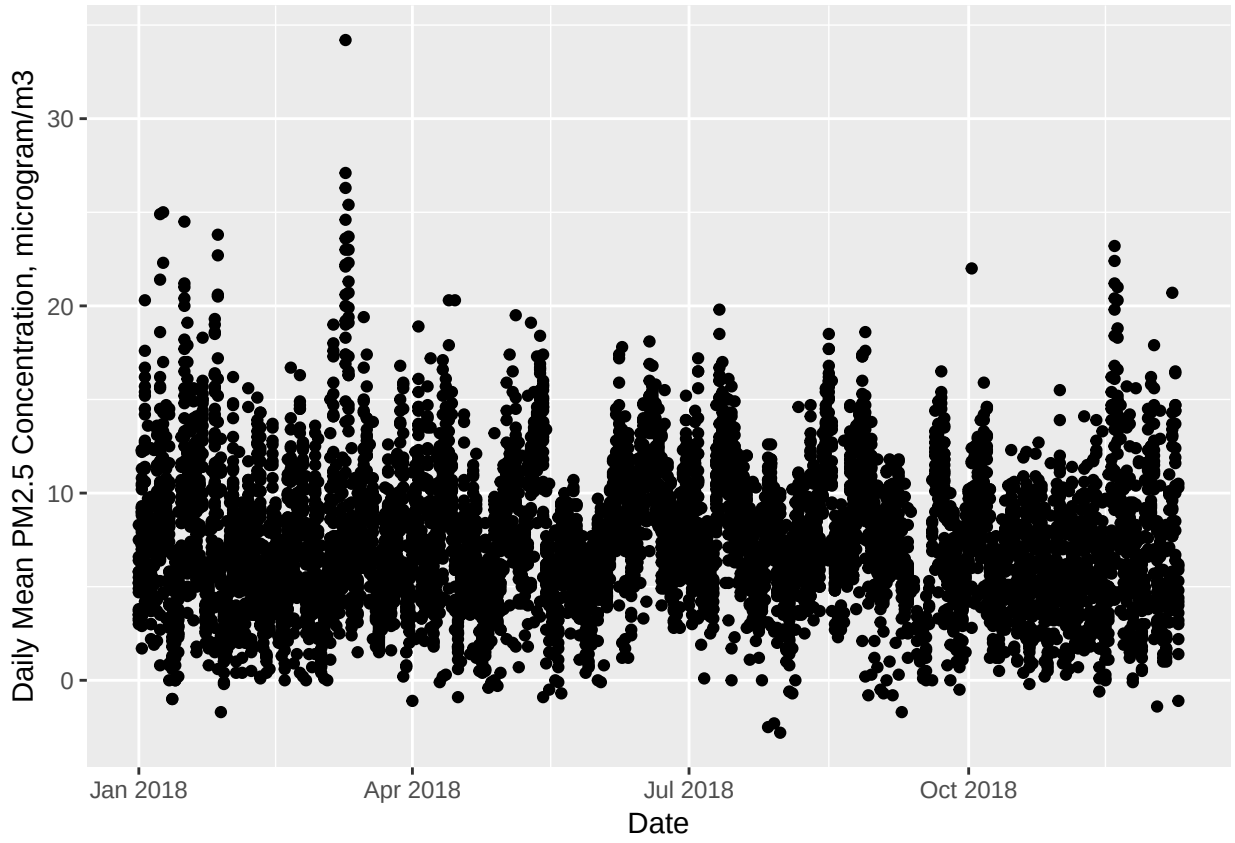
3a. Illustrate PM2.5 concentrations by date. Do not split aesthetics by site.

```
PMTest.mixed <- lme(data = EPAair.2018,
                     Daily.Mean.PM2.5.Concentration ~ Date,
                    random = ~1|Site.Name, method = "REML")
summary(PMTest.mixed)
```

```
## Linear mixed-effects model fit by REML
##  Data: EPAair.2018
##        AIC      BIC    logLik
##   40602.76 40630.51 -20297.38
##
## Random effects:
##  Formula: ~1 | Site.Name
##         (Intercept) Residual
## StdDev:    1.841425 3.457061
##
## Fixed effects: Daily.Mean.PM2.5.Concentration ~ Date
##                 Value Std.Error   DF   t-value p-value
## (Intercept) 20.141836  7.382570 7586  2.728296  0.0064
## Date        -0.000742  0.000417 7586 -1.779991  0.0751
##  Correlation:
##      (Intr)
## Date -0.999
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max
```

```
## -3.4251256 -0.6846871 -0.1385351  0.5919707  7.9199389
##
## Number of Observations: 7611
## Number of Groups: 24
```

```
ggplot(EPAair.2018, aes(x = Date, y = Daily.Mean.PM2.5.Concentration)) +
  geom_point() +
  labs(x = "Date", y = "Daily Mean PM2.5 Concentration, microgram/m3")
```



3b. Insert the following line of code into your R chunk. This will eliminate duplicate measurements on single dates for each site. PM2.5 = PM2.5[order(PM2.5[,'Date'],-PM2.5[,'Site.ID']),] PM2.5 = PM2.5[!duplicated(PM2.5$Date),]

3c. Determine the temporal autocorrelation in your model.

3d. Run a mixed effects model.

```
#PM2.5 = PM2.5[order(PM2.5[,'Date'],-PM2.5[,'Site.ID']),]
#PM2.5 = PM2.5[!duplicated(PM2.5$Date),]

PMTest.auto <- lme(data = EPAair.2018,
                   Daily.Mean.PM2.5.Concentration ~ Date,
                   random = ~1|Site.Name,  method = "REML")
PMTest.auto
```

```
## Linear mixed-effects model fit by REML
##   Data: EPAair.2018
##   Log-restricted-likelihood: -20297.38
##   Fixed: Daily.Mean.PM2.5.Concentration ~ Date
```

```
## (Intercept)        Date
## 20.14183588 -0.00074241
##
## Random effects:
##  Formula: ~1 | Site.Name
##         (Intercept) Residual
## StdDev:    1.841425 3.457061
##
## Number of Observations: 7611
## Number of Groups: 24
```

```
ACF(PMTest.auto)
```

```
##    lag          ACF
## 1    0  1.000000000
## 2    1  0.473017989
## 3    2  0.143093030
## 4    3  0.060500838
## 5    4  0.061574447
## 6    5  0.087756109
## 7    6  0.061116723
## 8    7  0.007595491
## 9    8  0.025491472
## 10   9  0.057872193
## 11  10  0.095911195
## 12  11  0.086519308
## 13  12  0.041507759
## 14  13  0.041091743
## 15  14  0.008663124
## 16  15 -0.012810524
## 17  16 -0.016388970
## 18  17 -0.023436707
## 19  18  0.020967717
## 20  19  0.032373855
## 21  20 -0.046770645
## 22  21 -0.086974675
## 23  22 -0.045009633
## 24  23  0.014507171
## 25  24  0.046279402
## 26  25  0.021031653
## 27  26 -0.017185250
## 28  27  0.008158717
```

```
PMTest..mixed <- lme(data = EPAair.2018,
                     Daily.Mean.PM2.5.Concentration ~ Date,
                     random = ~1|Site.Name,
#                      correlation =
#                      corAR1(form = ~ Date|Site.Name, value = 0.473),
                     method = "REML")
summary(PMTest..mixed)
```

```
## Linear mixed-effects model fit by REML
##  Data: EPAair.2018
##        AIC      BIC    logLik
##   40602.76 40630.51 -20297.38
```

```
##
## Random effects:
##  Formula: ~1 | Site.Name
##         (Intercept) Residual
## StdDev:    1.841425 3.457061
##
## Fixed effects: Daily.Mean.PM2.5.Concentration ~ Date
##                 Value Std.Error   DF   t-value p-value
## (Intercept) 20.141836  7.382570 7586  2.728296  0.0064
## Date        -0.000742  0.000417 7586 -1.779991  0.0751
##  Correlation:
##      (Intr)
## Date -0.999
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max
## -3.4251256 -0.6846871 -0.1385351  0.5919707  7.9199389
##
## Number of Observations: 7611
## Number of Groups: 24
```

Is there a significant increasing or decreasing trend in PM2.5 concentrations in 2018?

ANSWER: No.

3e. Run a fixed effects model with Date as the only explanatory variable. Then test whether the mixed effects model is a better fit than the fixed effect model.

```
PMTest.fixed <- gls(data = EPAair.2018,
                    Daily.Mean.PM2.5.Concentration ~ Date,
                    method = "REML")
summary(PMTest.fixed)
```

```
## Generalized least squares fit by REML
##   Model: Daily.Mean.PM2.5.Concentration ~ Date
##   Data: EPAair.2018
##       AIC      BIC    logLik
##   41493.9 41514.71 -20743.95
##
## Coefficients:
##                 Value Std.Error   t-value p-value
## (Intercept) 18.382226  7.761784  2.368299  0.0179
## Date        -0.000612  0.000439 -1.395100  0.1630
##
##  Correlation:
##      (Intr)
## Date -1
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.7977378 -0.6903373 -0.1103274  0.6016182  7.2066598
##
## Residual standard error: 3.689625
## Degrees of freedom: 7611 total; 7609 residual
```

```
anova(PMTest.mixed, PMTest.fixed)
```

```
##               Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## PMTest.mixed      1  4 40602.76 40630.51 -20297.38
## PMTest.fixed      2  3 41493.90 41514.71 -20743.95 1 vs 2 893.1307  <.0001
```

Which model is better?

ANSWER: Mixed effect model is better as it has a lower AIC value.

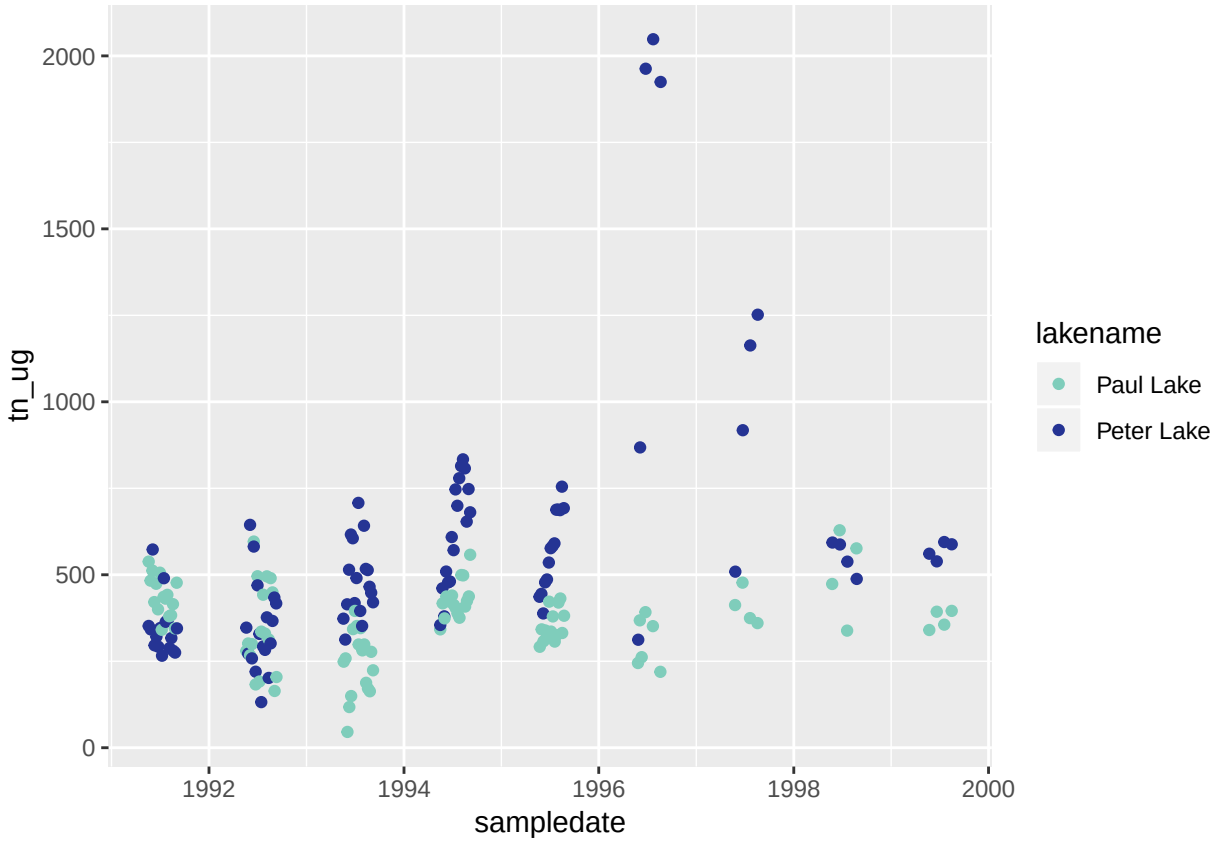
Run a Mann-Kendall test

Research question: Is there a trend in total N surface concentrations in Peter and Paul lakes?

4. Duplicate the Mann-Kendall test we ran for total P in class, this time with total N for both lakes. Make sure to run a test for changepoints in the datasets (and run a second one if a second change point is likely).

```
NTLnutrients.surface <-
  NTLnutrients %>%
  select(-lakeid, -depth_id, -comments) %>%
  filter(depth == 0) %>%
  filter(!is.na(tn_ug))

# Initial visualization of data
ggplot(NTLnutrients.surface, aes(x = sampledate, y = tn_ug, color = lakename)) +
  geom_point() +
  scale_color_manual(values = c("#7fcdbb", "#253494"))
```




```
# Split dataset by lake
Peter.nutrients.surface <- filter(NTLnutrients.surface, lakename == "Peter Lake")
Paul.nutrients.surface <- filter(NTLnutrients.surface, lakename == "Paul Lake")

#Mann-Kendall test
mk.test(Peter.nutrients.surface$tn_ug)
```

```
##
##  Mann-Kendall trend test
##
## data:  Peter.nutrients.surface$tn_ug
## z = 7.2927, n = 98, p-value = 3.039e-13
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## 2.377000e+03 1.061503e+05 5.001052e-01
```

```
# Pettitt Test
pettitt.test(Peter.nutrients.surface$tn_ug)
```

```
##
##  Pettitt's test for single change-point detection
##
## data:  Peter.nutrients.surface$tn_ug
## U* = 1884, p-value = 3.744e-10
## alternative hypothesis: two.sided
## sample estimates:
```

```
## probable change point at time K
##                              36
```

```
# Run separate Mann-Kendall for each change point
mk.test(Peter.nutrients.surface$tn_ug[1:37])
```

```
##
##  Mann-Kendall trend test
##
## data:  Peter.nutrients.surface$tn_ug[1:37]
## z = 0.4316, n = 37, p-value = 0.666
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## 3.400000e+01 5.846000e+03 5.105105e-02
```

```
mk.test(Peter.nutrients.surface$tn_ug[38:98])
```

```
##
##  Mann-Kendall trend test
##
## data:  Peter.nutrients.surface$tn_ug[38:98]
## z = 2.931, n = 61, p-value = 0.003379
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## 4.720000e+02 2.582333e+04 2.579235e-01
```

```
# Is there a second change point?
pettitt.test(Peter.nutrients.surface$tp_ug[38:98])
```

```
##
##  Pettitt's test for single change-point detection
##
## data:  Peter.nutrients.surface$tp_ug[38:98]
## U* = 378, p-value = 0.04866
## alternative hypothesis: two.sided
## sample estimates:
## probable change point at time K
##                              53
```

```
# Run another Mann-Kendall for the second change point
mk.test(Peter.nutrients.surface$tn_ug[1:37])
```

```
##
##  Mann-Kendall trend test
##
## data:  Peter.nutrients.surface$tn_ug[1:37]
## z = 0.4316, n = 37, p-value = 0.666
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## 3.400000e+01 5.846000e+03 5.105105e-02
```

```
mk.test(Peter.nutrients.surface$tp_ug[38:54])
```

```
##
##  Mann-Kendall trend test
```

```
##
## data:  Peter.nutrients.surface$tp_ug[38:54]
## z = -4.4076, n = 17, p-value = 1.045e-05
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## -108.0000000  589.3333333   -0.7941176
```

```
mk.test(Peter.nutrients.surface$tp_ug[55:98])
```

```
##
##  Mann-Kendall trend test
##
## data:  Peter.nutrients.surface$tp_ug[55:98]
## z = -1.2643, n = 44, p-value = 0.2061
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## -126.0000000 9775.3333333   -0.1331924
```

```
# Run the same test for Paul Lake
mk.test(Paul.nutrients.surface$tn_ug)
```

```
##
##  Mann-Kendall trend test
##
## data:  Paul.nutrients.surface$tn_ug
## z = -0.35068, n = 99, p-value = 0.7258
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -1.170000e+02  1.094170e+05 -2.411874e-02
```

```
pettitt.test(Paul.nutrients.surface$tn_ug)
```

```
##
##  Pettitt's test for single change-point detection
##
## data:  Paul.nutrients.surface$tn_ug
## U* = 704, p-value = 0.09624
## alternative hypothesis: two.sided
## sample estimates:
## probable change point at time K
##                              16
```

What are the results of this test?

> ANSWER: For Peter Lake: The Pettitt Test showed two change points needing to be taken into account. After doing so, the separate Mann-Kendall test results show negative z-scores indiciating there is a negative trend amongst the data. For only the middle portion of the data is there a monotonic trend, and for the beginning and end portions there is not as indicated by the p-values > 0.05. For Paul Lake: The low, negative z-score associated with the test indiciates there is a slight negative trend amongst the data. The p-value > 0.05 indicates we should not accept the alternative hypothesis that there is a monotonic trend in the data (z= -0.35068, p-value = 0.7258). The Pettitt Test also has non-significant p-value indicating there is no significant change point to needed to be taken into account (p-value = 0.09624).

5. Generate a graph that illustrates the TN concentrations over time, coloring by lake and adding vertical

line(s) representing changepoint(s).

```
# Add vertical lines to the original graph to represent change points
ggplot(NTLnutrients.surface, aes(x = sampledate, y = tn_ug, color = lakename)) +
  geom_point() +
  scale_color_manual(values = c("#7fcdbb", "#253494")) +
  geom_vline(xintercept = as.Date("1993-06-02"), linetype="dashed", color="#253494", size=1) +
  geom_vline(xintercept = as.Date("1994-06-01"), linetype="dashed", color="#253494", size=1) +
  labs(x = "Sample Date", y = "Total Nitrogen, micrograms", color = "Lake Name")
```