Assignment 7: Time Series Analysis

Katherine Owens

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

setwd("C:/Users/Katherine/Documents/872-Data Analytics/Environmental_Data_Analytics_2022")
getwd()
```

```
## [1] "C:/Users/Katherine/Documents/872-Data Analytics/Environmental_Data_Analytics_2022"
```

```
library(lubridate)
library(tidyverse)
library(dplyr)
#install.packages("trend")
library(trend)
#install.packages("zoo")
library(zoo)
#install.packages("Kendall")
library(Kendall)
#install.packages("tseries")
library(tseries)

#set ggplot theme
mytheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "purple"),
        legend.position = "bottom")
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.

   Question:how to do this all at one time?

```
#2

#Read Ozone-NC data
O3_10 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",stringsAsFactors = TRUE

O3_11 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",stringsAsFactors = TRUE

O3_12 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",stringsAsFactors = TRUE

O3_13 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",stringsAsFactors = TRUE

O3_14 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",stringsAsFactors = TRUE

O3_15 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",stringsAsFactors = TRUE

O3_16 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",stringsAsFactors = TRUE

O3_17 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",stringsAsFactors = TRUE

O3_18 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",stringsAsFactors = TRUE

O3_19 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",stringsAsFactors = TRUE

#Combine into aggregated df
GaringerOzone <- rbind(O3_10, O3_11, O3_12, O3_13, O3_14, O3_15, O3_16, O3_17, O3_18, O3_19)

#to check is a df
is.data.frame(GaringerOzone)
```

```
## [1] TRUE
```

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

#Format Date
```

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

#4

#Make df with only columns Date, Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE.
GOZ_3 <- select(GaringerOzone, Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

#5

#create new daily df with all days from '10-'19
?seq.Date
```

```
## starting httpd help server ... done
```

```r
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), "day"))

#rename single column in Days to Date
colnames(Days) <- c("Date")

#6

#combine df with missing days with daily df
GaringerOzone <-
  left_join(
  Days,
  GOZ_3)
```

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

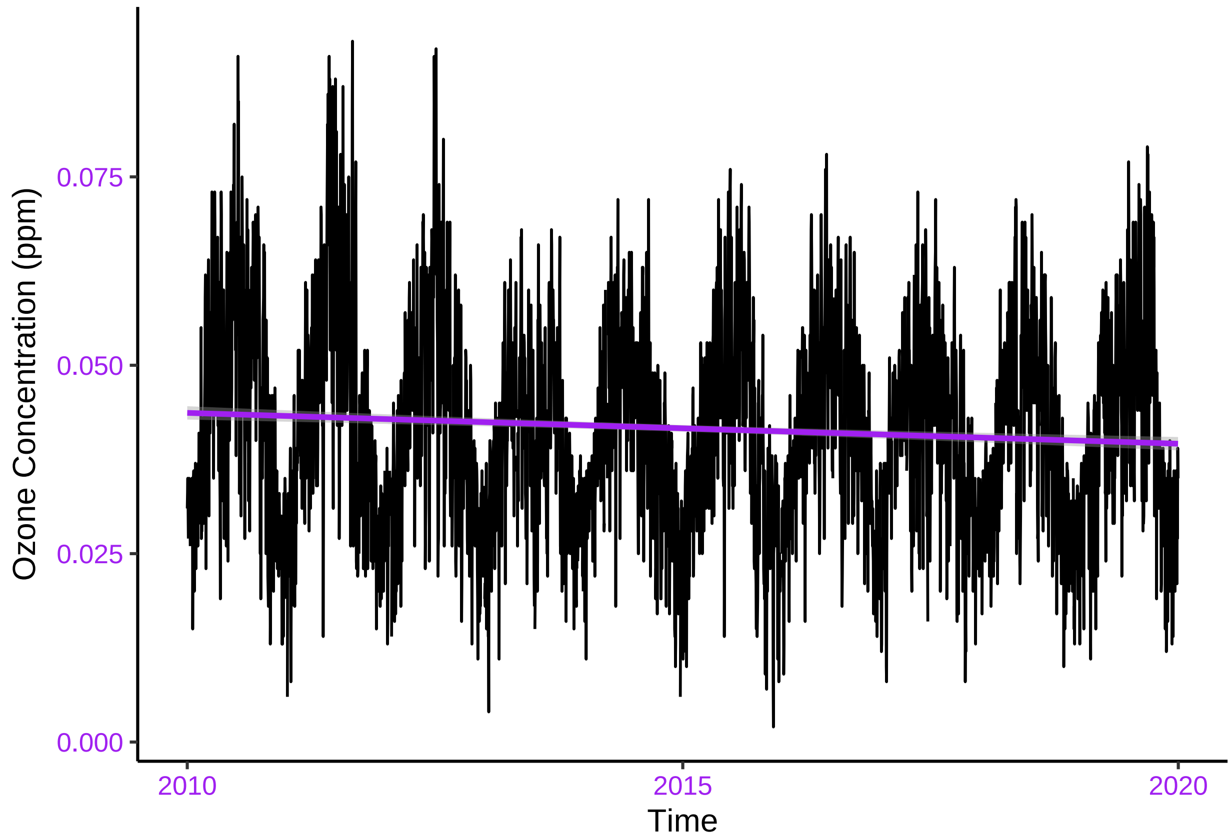
Visualize

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?

```r
#7

OzoneLine.plot <- ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) + #Pic
  geom_line() +
  labs(x = "Time", y = expression("Ozone Concentration (ppm)")) +
  geom_smooth(method = lm, color = "purple") #add a trendline
print(OzoneLine.plot)
```

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

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



Answer: Yes there appears to be a seasonality trend, as well as a slightly decreasing ozone concentration trend over time. The concentrations used to be more extreme with the swings between seasons being higher and lower with a range of 0.010-.095ppm. Later in the 2010s the ppm concentration range decreased to .015-0.075 ozone ppm range

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

summary(GaringerOzone) #63 NAs
```

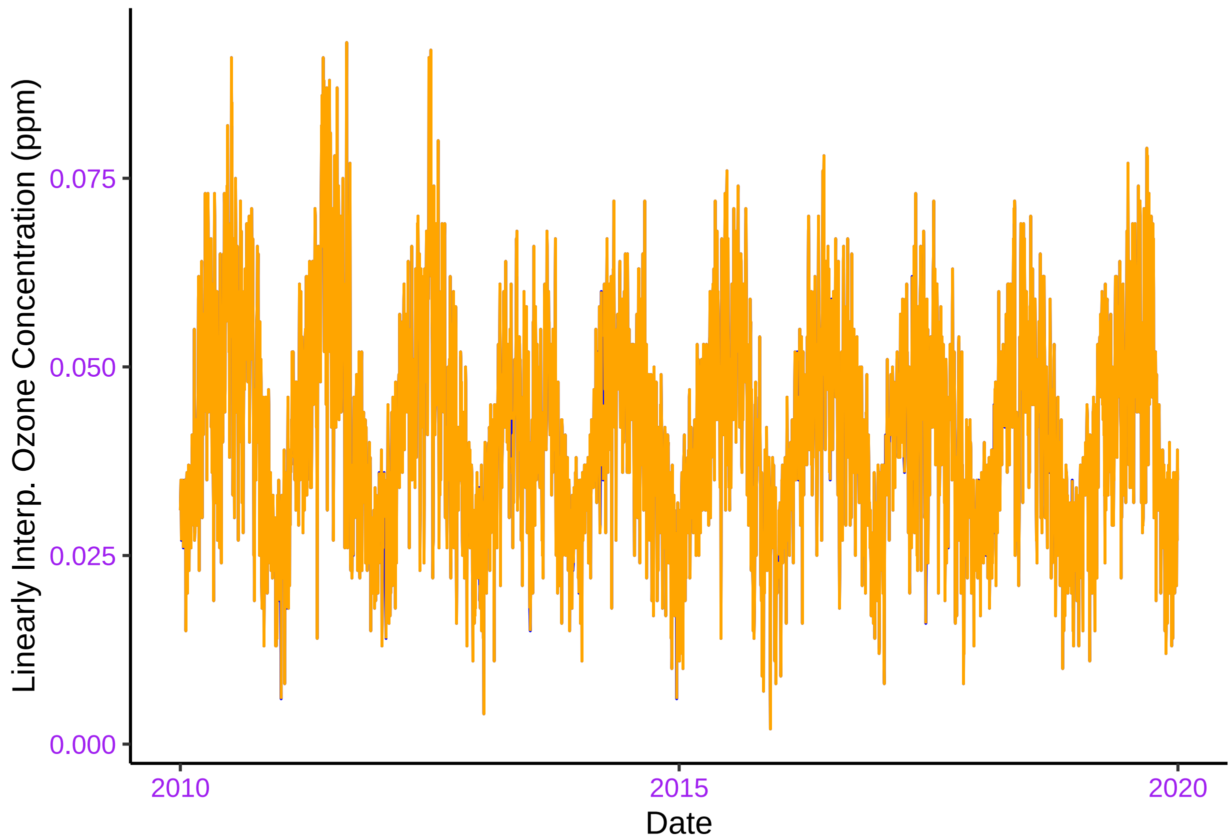
```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
##                       NA's   :63                           NA's   :63
```

```
GaringerOzone_clean <- #removing NAs
  GaringerOzone %>%  #referencing data frame to use
  mutate(Daily.Max.8.hour.Ozone.Concentration_clean = #making new clean column
```

```
        zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration)) #cut NAs

summary(GaringerOzone_clean$Daily.Max.8.hour.Ozone.Concentration_clean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00200 0.03200 0.04100 0.04151 0.05100 0.09300
```

```
#line plot of interpolated data
ggplot(GaringerOzone_clean) +
  geom_line(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration_clean), color = "blue") +
  geom_line(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration), color = "orange") +
  ylab("Linearly Interp. Ozone Concentration (ppm)")
```



Answer: We didn't use piecewise because it would not help with line aspect needed for time series to bridge a gab in data by only providing the nearest neighbor option. We didn't use a spline interpolation because it would add a curved, quadratic line when we only need a straight line in this case to connect the data. Probably better in other uses, but for this exercise the linear interpolation satisfied the linear data need well.

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 (have questions)

GaringerOzone_monthly <-
  GaringerOzone_clean %>% #creating new df
```

```
  mutate(Month = month(Date), #forming new columns(m&yr)
         Year = year(Date)) %>%
  mutate(Date = my(paste0(Month, "-", Year))) %>% #creating new Date column (month-year) combos
  dplyr::group_by(Date, Month, Year) %>%
  dplyr::summarise (Mean_Ozone = mean(Daily.Max.8.hour.Ozone.Concentration_clean)) %>%
  select(Mean_Ozone, Date)
```

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

## Adding missing grouping variables: `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

#daily set
f_day <- day(first(GaringerOzone_clean$Date))
f_month <- month(first(GaringerOzone_clean$Date))
f_year <- year(first(GaringerOzone_clean$Date))

GaringerOzone.daily.ts <- ts(GaringerOzone_clean$Date, start = c(f_year, f_month, f_day), frequency=365
print(GaringerOzone.daily.ts)
```

```
## Time Series:
## Start = c(2010, 1)
## End = c(2020, 2)
## Frequency = 365
##    [1] 14610 14611 14612 14613 14614 14615 14616 14617 14618 14619 14620 14621
##   [13] 14622 14623 14624 14625 14626 14627 14628 14629 14630 14631 14632 14633
##   [25] 14634 14635 14636 14637 14638 14639 14640 14641 14642 14643 14644 14645
##   [37] 14646 14647 14648 14649 14650 14651 14652 14653 14654 14655 14656 14657
##   [49] 14658 14659 14660 14661 14662 14663 14664 14665 14666 14667 14668 14669
##   [61] 14670 14671 14672 14673 14674 14675 14676 14677 14678 14679 14680 14681
##   [73] 14682 14683 14684 14685 14686 14687 14688 14689 14690 14691 14692 14693
##   [85] 14694 14695 14696 14697 14698 14699 14700 14701 14702 14703 14704 14705
##   [97] 14706 14707 14708 14709 14710 14711 14712 14713 14714 14715 14716 14717
##  [109] 14718 14719 14720 14721 14722 14723 14724 14725 14726 14727 14728 14729
##  [121] 14730 14731 14732 14733 14734 14735 14736 14737 14738 14739 14740 14741
##  [133] 14742 14743 14744 14745 14746 14747 14748 14749 14750 14751 14752 14753
##  [145] 14754 14755 14756 14757 14758 14759 14760 14761 14762 14763 14764 14765
##  [157] 14766 14767 14768 14769 14770 14771 14772 14773 14774 14775 14776 14777
##  [169] 14778 14779 14780 14781 14782 14783 14784 14785 14786 14787 14788 14789
##  [181] 14790 14791 14792 14793 14794 14795 14796 14797 14798 14799 14800 14801
##  [193] 14802 14803 14804 14805 14806 14807 14808 14809 14810 14811 14812 14813
##  [205] 14814 14815 14816 14817 14818 14819 14820 14821 14822 14823 14824 14825
##  [217] 14826 14827 14828 14829 14830 14831 14832 14833 14834 14835 14836 14837
##  [229] 14838 14839 14840 14841 14842 14843 14844 14845 14846 14847 14848 14849
##  [241] 14850 14851 14852 14853 14854 14855 14856 14857 14858 14859 14860 14861
##  [253] 14862 14863 14864 14865 14866 14867 14868 14869 14870 14871 14872 14873
##  [265] 14874 14875 14876 14877 14878 14879 14880 14881 14882 14883 14884 14885
##  [277] 14886 14887 14888 14889 14890 14891 14892 14893 14894 14895 14896 14897
##  [289] 14898 14899 14900 14901 14902 14903 14904 14905 14906 14907 14908 14909
```

```
## [301] 14910 14911 14912 14913 14914 14915 14916 14917 14918 14919 14920 14921
## [313] 14922 14923 14924 14925 14926 14927 14928 14929 14930 14931 14932 14933
## [325] 14934 14935 14936 14937 14938 14939 14940 14941 14942 14943 14944 14945
## [337] 14946 14947 14948 14949 14950 14951 14952 14953 14954 14955 14956 14957
## [349] 14958 14959 14960 14961 14962 14963 14964 14965 14966 14967 14968 14969
## [361] 14970 14971 14972 14973 14974 14975 14976 14977 14978 14979 14980 14981
## [373] 14982 14983 14984 14985 14986 14987 14988 14989 14990 14991 14992 14993
## [385] 14994 14995 14996 14997 14998 14999 15000 15001 15002 15003 15004 15005
## [397] 15006 15007 15008 15009 15010 15011 15012 15013 15014 15015 15016 15017
## [409] 15018 15019 15020 15021 15022 15023 15024 15025 15026 15027 15028 15029
## [421] 15030 15031 15032 15033 15034 15035 15036 15037 15038 15039 15040 15041
## [433] 15042 15043 15044 15045 15046 15047 15048 15049 15050 15051 15052 15053
## [445] 15054 15055 15056 15057 15058 15059 15060 15061 15062 15063 15064 15065
## [457] 15066 15067 15068 15069 15070 15071 15072 15073 15074 15075 15076 15077
## [469] 15078 15079 15080 15081 15082 15083 15084 15085 15086 15087 15088 15089
## [481] 15090 15091 15092 15093 15094 15095 15096 15097 15098 15099 15100 15101
## [493] 15102 15103 15104 15105 15106 15107 15108 15109 15110 15111 15112 15113
## [505] 15114 15115 15116 15117 15118 15119 15120 15121 15122 15123 15124 15125
## [517] 15126 15127 15128 15129 15130 15131 15132 15133 15134 15135 15136 15137
## [529] 15138 15139 15140 15141 15142 15143 15144 15145 15146 15147 15148 15149
## [541] 15150 15151 15152 15153 15154 15155 15156 15157 15158 15159 15160 15161
## [553] 15162 15163 15164 15165 15166 15167 15168 15169 15170 15171 15172 15173
## [565] 15174 15175 15176 15177 15178 15179 15180 15181 15182 15183 15184 15185
## [577] 15186 15187 15188 15189 15190 15191 15192 15193 15194 15195 15196 15197
## [589] 15198 15199 15200 15201 15202 15203 15204 15205 15206 15207 15208 15209
## [601] 15210 15211 15212 15213 15214 15215 15216 15217 15218 15219 15220 15221
## [613] 15222 15223 15224 15225 15226 15227 15228 15229 15230 15231 15232 15233
## [625] 15234 15235 15236 15237 15238 15239 15240 15241 15242 15243 15244 15245
## [637] 15246 15247 15248 15249 15250 15251 15252 15253 15254 15255 15256 15257
## [649] 15258 15259 15260 15261 15262 15263 15264 15265 15266 15267 15268 15269
## [661] 15270 15271 15272 15273 15274 15275 15276 15277 15278 15279 15280 15281
## [673] 15282 15283 15284 15285 15286 15287 15288 15289 15290 15291 15292 15293
## [685] 15294 15295 15296 15297 15298 15299 15300 15301 15302 15303 15304 15305
## [697] 15306 15307 15308 15309 15310 15311 15312 15313 15314 15315 15316 15317
## [709] 15318 15319 15320 15321 15322 15323 15324 15325 15326 15327 15328 15329
## [721] 15330 15331 15332 15333 15334 15335 15336 15337 15338 15339 15340 15341
## [733] 15342 15343 15344 15345 15346 15347 15348 15349 15350 15351 15352 15353
## [745] 15354 15355 15356 15357 15358 15359 15360 15361 15362 15363 15364 15365
## [757] 15366 15367 15368 15369 15370 15371 15372 15373 15374 15375 15376 15377
## [769] 15378 15379 15380 15381 15382 15383 15384 15385 15386 15387 15388 15389
## [781] 15390 15391 15392 15393 15394 15395 15396 15397 15398 15399 15400 15401
## [793] 15402 15403 15404 15405 15406 15407 15408 15409 15410 15411 15412 15413
## [805] 15414 15415 15416 15417 15418 15419 15420 15421 15422 15423 15424 15425
## [817] 15426 15427 15428 15429 15430 15431 15432 15433 15434 15435 15436 15437
## [829] 15438 15439 15440 15441 15442 15443 15444 15445 15446 15447 15448 15449
## [841] 15450 15451 15452 15453 15454 15455 15456 15457 15458 15459 15460 15461
## [853] 15462 15463 15464 15465 15466 15467 15468 15469 15470 15471 15472 15473
## [865] 15474 15475 15476 15477 15478 15479 15480 15481 15482 15483 15484 15485
## [877] 15486 15487 15488 15489 15490 15491 15492 15493 15494 15495 15496 15497
## [889] 15498 15499 15500 15501 15502 15503 15504 15505 15506 15507 15508 15509
## [901] 15510 15511 15512 15513 15514 15515 15516 15517 15518 15519 15520 15521
## [913] 15522 15523 15524 15525 15526 15527 15528 15529 15530 15531 15532 15533
## [925] 15534 15535 15536 15537 15538 15539 15540 15541 15542 15543 15544 15545
## [937] 15546 15547 15548 15549 15550 15551 15552 15553 15554 15555 15556 15557
```

```
##  [949] 15558 15559 15560 15561 15562 15563 15564 15565 15566 15567 15568 15569
##  [961] 15570 15571 15572 15573 15574 15575 15576 15577 15578 15579 15580 15581
##  [973] 15582 15583 15584 15585 15586 15587 15588 15589 15590 15591 15592 15593
##  [985] 15594 15595 15596 15597 15598 15599 15600 15601 15602 15603 15604 15605
##  [997] 15606 15607 15608 15609 15610 15611 15612 15613 15614 15615 15616 15617
## [1009] 15618 15619 15620 15621 15622 15623 15624 15625 15626 15627 15628 15629
## [1021] 15630 15631 15632 15633 15634 15635 15636 15637 15638 15639 15640 15641
## [1033] 15642 15643 15644 15645 15646 15647 15648 15649 15650 15651 15652 15653
## [1045] 15654 15655 15656 15657 15658 15659 15660 15661 15662 15663 15664 15665
## [1057] 15666 15667 15668 15669 15670 15671 15672 15673 15674 15675 15676 15677
## [1069] 15678 15679 15680 15681 15682 15683 15684 15685 15686 15687 15688 15689
## [1081] 15690 15691 15692 15693 15694 15695 15696 15697 15698 15699 15700 15701
## [1093] 15702 15703 15704 15705 15706 15707 15708 15709 15710 15711 15712 15713
## [1105] 15714 15715 15716 15717 15718 15719 15720 15721 15722 15723 15724 15725
## [1117] 15726 15727 15728 15729 15730 15731 15732 15733 15734 15735 15736 15737
## [1129] 15738 15739 15740 15741 15742 15743 15744 15745 15746 15747 15748 15749
## [1141] 15750 15751 15752 15753 15754 15755 15756 15757 15758 15759 15760 15761
## [1153] 15762 15763 15764 15765 15766 15767 15768 15769 15770 15771 15772 15773
## [1165] 15774 15775 15776 15777 15778 15779 15780 15781 15782 15783 15784 15785
## [1177] 15786 15787 15788 15789 15790 15791 15792 15793 15794 15795 15796 15797
## [1189] 15798 15799 15800 15801 15802 15803 15804 15805 15806 15807 15808 15809
## [1201] 15810 15811 15812 15813 15814 15815 15816 15817 15818 15819 15820 15821
## [1213] 15822 15823 15824 15825 15826 15827 15828 15829 15830 15831 15832 15833
## [1225] 15834 15835 15836 15837 15838 15839 15840 15841 15842 15843 15844 15845
## [1237] 15846 15847 15848 15849 15850 15851 15852 15853 15854 15855 15856 15857
## [1249] 15858 15859 15860 15861 15862 15863 15864 15865 15866 15867 15868 15869
## [1261] 15870 15871 15872 15873 15874 15875 15876 15877 15878 15879 15880 15881
## [1273] 15882 15883 15884 15885 15886 15887 15888 15889 15890 15891 15892 15893
## [1285] 15894 15895 15896 15897 15898 15899 15900 15901 15902 15903 15904 15905
## [1297] 15906 15907 15908 15909 15910 15911 15912 15913 15914 15915 15916 15917
## [1309] 15918 15919 15920 15921 15922 15923 15924 15925 15926 15927 15928 15929
## [1321] 15930 15931 15932 15933 15934 15935 15936 15937 15938 15939 15940 15941
## [1333] 15942 15943 15944 15945 15946 15947 15948 15949 15950 15951 15952 15953
## [1345] 15954 15955 15956 15957 15958 15959 15960 15961 15962 15963 15964 15965
## [1357] 15966 15967 15968 15969 15970 15971 15972 15973 15974 15975 15976 15977
## [1369] 15978 15979 15980 15981 15982 15983 15984 15985 15986 15987 15988 15989
## [1381] 15990 15991 15992 15993 15994 15995 15996 15997 15998 15999 16000 16001
## [1393] 16002 16003 16004 16005 16006 16007 16008 16009 16010 16011 16012 16013
## [1405] 16014 16015 16016 16017 16018 16019 16020 16021 16022 16023 16024 16025
## [1417] 16026 16027 16028 16029 16030 16031 16032 16033 16034 16035 16036 16037
## [1429] 16038 16039 16040 16041 16042 16043 16044 16045 16046 16047 16048 16049
## [1441] 16050 16051 16052 16053 16054 16055 16056 16057 16058 16059 16060 16061
## [1453] 16062 16063 16064 16065 16066 16067 16068 16069 16070 16071 16072 16073
## [1465] 16074 16075 16076 16077 16078 16079 16080 16081 16082 16083 16084 16085
## [1477] 16086 16087 16088 16089 16090 16091 16092 16093 16094 16095 16096 16097
## [1489] 16098 16099 16100 16101 16102 16103 16104 16105 16106 16107 16108 16109
## [1501] 16110 16111 16112 16113 16114 16115 16116 16117 16118 16119 16120 16121
## [1513] 16122 16123 16124 16125 16126 16127 16128 16129 16130 16131 16132 16133
## [1525] 16134 16135 16136 16137 16138 16139 16140 16141 16142 16143 16144 16145
## [1537] 16146 16147 16148 16149 16150 16151 16152 16153 16154 16155 16156 16157
## [1549] 16158 16159 16160 16161 16162 16163 16164 16165 16166 16167 16168 16169
## [1561] 16170 16171 16172 16173 16174 16175 16176 16177 16178 16179 16180 16181
## [1573] 16182 16183 16184 16185 16186 16187 16188 16189 16190 16191 16192 16193
## [1585] 16194 16195 16196 16197 16198 16199 16200 16201 16202 16203 16204 16205
```

```
## [1597] 16206 16207 16208 16209 16210 16211 16212 16213 16214 16215 16216 16217
## [1609] 16218 16219 16220 16221 16222 16223 16224 16225 16226 16227 16228 16229
## [1621] 16230 16231 16232 16233 16234 16235 16236 16237 16238 16239 16240 16241
## [1633] 16242 16243 16244 16245 16246 16247 16248 16249 16250 16251 16252 16253
## [1645] 16254 16255 16256 16257 16258 16259 16260 16261 16262 16263 16264 16265
## [1657] 16266 16267 16268 16269 16270 16271 16272 16273 16274 16275 16276 16277
## [1669] 16278 16279 16280 16281 16282 16283 16284 16285 16286 16287 16288 16289
## [1681] 16290 16291 16292 16293 16294 16295 16296 16297 16298 16299 16300 16301
## [1693] 16302 16303 16304 16305 16306 16307 16308 16309 16310 16311 16312 16313
## [1705] 16314 16315 16316 16317 16318 16319 16320 16321 16322 16323 16324 16325
## [1717] 16326 16327 16328 16329 16330 16331 16332 16333 16334 16335 16336 16337
## [1729] 16338 16339 16340 16341 16342 16343 16344 16345 16346 16347 16348 16349
## [1741] 16350 16351 16352 16353 16354 16355 16356 16357 16358 16359 16360 16361
## [1753] 16362 16363 16364 16365 16366 16367 16368 16369 16370 16371 16372 16373
## [1765] 16374 16375 16376 16377 16378 16379 16380 16381 16382 16383 16384 16385
## [1777] 16386 16387 16388 16389 16390 16391 16392 16393 16394 16395 16396 16397
## [1789] 16398 16399 16400 16401 16402 16403 16404 16405 16406 16407 16408 16409
## [1801] 16410 16411 16412 16413 16414 16415 16416 16417 16418 16419 16420 16421
## [1813] 16422 16423 16424 16425 16426 16427 16428 16429 16430 16431 16432 16433
## [1825] 16434 16435 16436 16437 16438 16439 16440 16441 16442 16443 16444 16445
## [1837] 16446 16447 16448 16449 16450 16451 16452 16453 16454 16455 16456 16457
## [1849] 16458 16459 16460 16461 16462 16463 16464 16465 16466 16467 16468 16469
## [1861] 16470 16471 16472 16473 16474 16475 16476 16477 16478 16479 16480 16481
## [1873] 16482 16483 16484 16485 16486 16487 16488 16489 16490 16491 16492 16493
## [1885] 16494 16495 16496 16497 16498 16499 16500 16501 16502 16503 16504 16505
## [1897] 16506 16507 16508 16509 16510 16511 16512 16513 16514 16515 16516 16517
## [1909] 16518 16519 16520 16521 16522 16523 16524 16525 16526 16527 16528 16529
## [1921] 16530 16531 16532 16533 16534 16535 16536 16537 16538 16539 16540 16541
## [1933] 16542 16543 16544 16545 16546 16547 16548 16549 16550 16551 16552 16553
## [1945] 16554 16555 16556 16557 16558 16559 16560 16561 16562 16563 16564 16565
## [1957] 16566 16567 16568 16569 16570 16571 16572 16573 16574 16575 16576 16577
## [1969] 16578 16579 16580 16581 16582 16583 16584 16585 16586 16587 16588 16589
## [1981] 16590 16591 16592 16593 16594 16595 16596 16597 16598 16599 16600 16601
## [1993] 16602 16603 16604 16605 16606 16607 16608 16609 16610 16611 16612 16613
## [2005] 16614 16615 16616 16617 16618 16619 16620 16621 16622 16623 16624 16625
## [2017] 16626 16627 16628 16629 16630 16631 16632 16633 16634 16635 16636 16637
## [2029] 16638 16639 16640 16641 16642 16643 16644 16645 16646 16647 16648 16649
## [2041] 16650 16651 16652 16653 16654 16655 16656 16657 16658 16659 16660 16661
## [2053] 16662 16663 16664 16665 16666 16667 16668 16669 16670 16671 16672 16673
## [2065] 16674 16675 16676 16677 16678 16679 16680 16681 16682 16683 16684 16685
## [2077] 16686 16687 16688 16689 16690 16691 16692 16693 16694 16695 16696 16697
## [2089] 16698 16699 16700 16701 16702 16703 16704 16705 16706 16707 16708 16709
## [2101] 16710 16711 16712 16713 16714 16715 16716 16717 16718 16719 16720 16721
## [2113] 16722 16723 16724 16725 16726 16727 16728 16729 16730 16731 16732 16733
## [2125] 16734 16735 16736 16737 16738 16739 16740 16741 16742 16743 16744 16745
## [2137] 16746 16747 16748 16749 16750 16751 16752 16753 16754 16755 16756 16757
## [2149] 16758 16759 16760 16761 16762 16763 16764 16765 16766 16767 16768 16769
## [2161] 16770 16771 16772 16773 16774 16775 16776 16777 16778 16779 16780 16781
## [2173] 16782 16783 16784 16785 16786 16787 16788 16789 16790 16791 16792 16793
## [2185] 16794 16795 16796 16797 16798 16799 16800 16801 16802 16803 16804 16805
## [2197] 16806 16807 16808 16809 16810 16811 16812 16813 16814 16815 16816 16817
## [2209] 16818 16819 16820 16821 16822 16823 16824 16825 16826 16827 16828 16829
## [2221] 16830 16831 16832 16833 16834 16835 16836 16837 16838 16839 16840 16841
## [2233] 16842 16843 16844 16845 16846 16847 16848 16849 16850 16851 16852 16853
```

```
## [2245] 16854 16855 16856 16857 16858 16859 16860 16861 16862 16863 16864 16865
## [2257] 16866 16867 16868 16869 16870 16871 16872 16873 16874 16875 16876 16877
## [2269] 16878 16879 16880 16881 16882 16883 16884 16885 16886 16887 16888 16889
## [2281] 16890 16891 16892 16893 16894 16895 16896 16897 16898 16899 16900 16901
## [2293] 16902 16903 16904 16905 16906 16907 16908 16909 16910 16911 16912 16913
## [2305] 16914 16915 16916 16917 16918 16919 16920 16921 16922 16923 16924 16925
## [2317] 16926 16927 16928 16929 16930 16931 16932 16933 16934 16935 16936 16937
## [2329] 16938 16939 16940 16941 16942 16943 16944 16945 16946 16947 16948 16949
## [2341] 16950 16951 16952 16953 16954 16955 16956 16957 16958 16959 16960 16961
## [2353] 16962 16963 16964 16965 16966 16967 16968 16969 16970 16971 16972 16973
## [2365] 16974 16975 16976 16977 16978 16979 16980 16981 16982 16983 16984 16985
## [2377] 16986 16987 16988 16989 16990 16991 16992 16993 16994 16995 16996 16997
## [2389] 16998 16999 17000 17001 17002 17003 17004 17005 17006 17007 17008 17009
## [2401] 17010 17011 17012 17013 17014 17015 17016 17017 17018 17019 17020 17021
## [2413] 17022 17023 17024 17025 17026 17027 17028 17029 17030 17031 17032 17033
## [2425] 17034 17035 17036 17037 17038 17039 17040 17041 17042 17043 17044 17045
## [2437] 17046 17047 17048 17049 17050 17051 17052 17053 17054 17055 17056 17057
## [2449] 17058 17059 17060 17061 17062 17063 17064 17065 17066 17067 17068 17069
## [2461] 17070 17071 17072 17073 17074 17075 17076 17077 17078 17079 17080 17081
## [2473] 17082 17083 17084 17085 17086 17087 17088 17089 17090 17091 17092 17093
## [2485] 17094 17095 17096 17097 17098 17099 17100 17101 17102 17103 17104 17105
## [2497] 17106 17107 17108 17109 17110 17111 17112 17113 17114 17115 17116 17117
## [2509] 17118 17119 17120 17121 17122 17123 17124 17125 17126 17127 17128 17129
## [2521] 17130 17131 17132 17133 17134 17135 17136 17137 17138 17139 17140 17141
## [2533] 17142 17143 17144 17145 17146 17147 17148 17149 17150 17151 17152 17153
## [2545] 17154 17155 17156 17157 17158 17159 17160 17161 17162 17163 17164 17165
## [2557] 17166 17167 17168 17169 17170 17171 17172 17173 17174 17175 17176 17177
## [2569] 17178 17179 17180 17181 17182 17183 17184 17185 17186 17187 17188 17189
## [2581] 17190 17191 17192 17193 17194 17195 17196 17197 17198 17199 17200 17201
## [2593] 17202 17203 17204 17205 17206 17207 17208 17209 17210 17211 17212 17213
## [2605] 17214 17215 17216 17217 17218 17219 17220 17221 17222 17223 17224 17225
## [2617] 17226 17227 17228 17229 17230 17231 17232 17233 17234 17235 17236 17237
## [2629] 17238 17239 17240 17241 17242 17243 17244 17245 17246 17247 17248 17249
## [2641] 17250 17251 17252 17253 17254 17255 17256 17257 17258 17259 17260 17261
## [2653] 17262 17263 17264 17265 17266 17267 17268 17269 17270 17271 17272 17273
## [2665] 17274 17275 17276 17277 17278 17279 17280 17281 17282 17283 17284 17285
## [2677] 17286 17287 17288 17289 17290 17291 17292 17293 17294 17295 17296 17297
## [2689] 17298 17299 17300 17301 17302 17303 17304 17305 17306 17307 17308 17309
## [2701] 17310 17311 17312 17313 17314 17315 17316 17317 17318 17319 17320 17321
## [2713] 17322 17323 17324 17325 17326 17327 17328 17329 17330 17331 17332 17333
## [2725] 17334 17335 17336 17337 17338 17339 17340 17341 17342 17343 17344 17345
## [2737] 17346 17347 17348 17349 17350 17351 17352 17353 17354 17355 17356 17357
## [2749] 17358 17359 17360 17361 17362 17363 17364 17365 17366 17367 17368 17369
## [2761] 17370 17371 17372 17373 17374 17375 17376 17377 17378 17379 17380 17381
## [2773] 17382 17383 17384 17385 17386 17387 17388 17389 17390 17391 17392 17393
## [2785] 17394 17395 17396 17397 17398 17399 17400 17401 17402 17403 17404 17405
## [2797] 17406 17407 17408 17409 17410 17411 17412 17413 17414 17415 17416 17417
## [2809] 17418 17419 17420 17421 17422 17423 17424 17425 17426 17427 17428 17429
## [2821] 17430 17431 17432 17433 17434 17435 17436 17437 17438 17439 17440 17441
## [2833] 17442 17443 17444 17445 17446 17447 17448 17449 17450 17451 17452 17453
## [2845] 17454 17455 17456 17457 17458 17459 17460 17461 17462 17463 17464 17465
## [2857] 17466 17467 17468 17469 17470 17471 17472 17473 17474 17475 17476 17477
## [2869] 17478 17479 17480 17481 17482 17483 17484 17485 17486 17487 17488 17489
## [2881] 17490 17491 17492 17493 17494 17495 17496 17497 17498 17499 17500 17501
```

```
## [2893] 17502 17503 17504 17505 17506 17507 17508 17509 17510 17511 17512 17513
## [2905] 17514 17515 17516 17517 17518 17519 17520 17521 17522 17523 17524 17525
## [2917] 17526 17527 17528 17529 17530 17531 17532 17533 17534 17535 17536 17537
## [2929] 17538 17539 17540 17541 17542 17543 17544 17545 17546 17547 17548 17549
## [2941] 17550 17551 17552 17553 17554 17555 17556 17557 17558 17559 17560 17561
## [2953] 17562 17563 17564 17565 17566 17567 17568 17569 17570 17571 17572 17573
## [2965] 17574 17575 17576 17577 17578 17579 17580 17581 17582 17583 17584 17585
## [2977] 17586 17587 17588 17589 17590 17591 17592 17593 17594 17595 17596 17597
## [2989] 17598 17599 17600 17601 17602 17603 17604 17605 17606 17607 17608 17609
## [3001] 17610 17611 17612 17613 17614 17615 17616 17617 17618 17619 17620 17621
## [3013] 17622 17623 17624 17625 17626 17627 17628 17629 17630 17631 17632 17633
## [3025] 17634 17635 17636 17637 17638 17639 17640 17641 17642 17643 17644 17645
## [3037] 17646 17647 17648 17649 17650 17651 17652 17653 17654 17655 17656 17657
## [3049] 17658 17659 17660 17661 17662 17663 17664 17665 17666 17667 17668 17669
## [3061] 17670 17671 17672 17673 17674 17675 17676 17677 17678 17679 17680 17681
## [3073] 17682 17683 17684 17685 17686 17687 17688 17689 17690 17691 17692 17693
## [3085] 17694 17695 17696 17697 17698 17699 17700 17701 17702 17703 17704 17705
## [3097] 17706 17707 17708 17709 17710 17711 17712 17713 17714 17715 17716 17717
## [3109] 17718 17719 17720 17721 17722 17723 17724 17725 17726 17727 17728 17729
## [3121] 17730 17731 17732 17733 17734 17735 17736 17737 17738 17739 17740 17741
## [3133] 17742 17743 17744 17745 17746 17747 17748 17749 17750 17751 17752 17753
## [3145] 17754 17755 17756 17757 17758 17759 17760 17761 17762 17763 17764 17765
## [3157] 17766 17767 17768 17769 17770 17771 17772 17773 17774 17775 17776 17777
## [3169] 17778 17779 17780 17781 17782 17783 17784 17785 17786 17787 17788 17789
## [3181] 17790 17791 17792 17793 17794 17795 17796 17797 17798 17799 17800 17801
## [3193] 17802 17803 17804 17805 17806 17807 17808 17809 17810 17811 17812 17813
## [3205] 17814 17815 17816 17817 17818 17819 17820 17821 17822 17823 17824 17825
## [3217] 17826 17827 17828 17829 17830 17831 17832 17833 17834 17835 17836 17837
## [3229] 17838 17839 17840 17841 17842 17843 17844 17845 17846 17847 17848 17849
## [3241] 17850 17851 17852 17853 17854 17855 17856 17857 17858 17859 17860 17861
## [3253] 17862 17863 17864 17865 17866 17867 17868 17869 17870 17871 17872 17873
## [3265] 17874 17875 17876 17877 17878 17879 17880 17881 17882 17883 17884 17885
## [3277] 17886 17887 17888 17889 17890 17891 17892 17893 17894 17895 17896 17897
## [3289] 17898 17899 17900 17901 17902 17903 17904 17905 17906 17907 17908 17909
## [3301] 17910 17911 17912 17913 17914 17915 17916 17917 17918 17919 17920 17921
## [3313] 17922 17923 17924 17925 17926 17927 17928 17929 17930 17931 17932 17933
## [3325] 17934 17935 17936 17937 17938 17939 17940 17941 17942 17943 17944 17945
## [3337] 17946 17947 17948 17949 17950 17951 17952 17953 17954 17955 17956 17957
## [3349] 17958 17959 17960 17961 17962 17963 17964 17965 17966 17967 17968 17969
## [3361] 17970 17971 17972 17973 17974 17975 17976 17977 17978 17979 17980 17981
## [3373] 17982 17983 17984 17985 17986 17987 17988 17989 17990 17991 17992 17993
## [3385] 17994 17995 17996 17997 17998 17999 18000 18001 18002 18003 18004 18005
## [3397] 18006 18007 18008 18009 18010 18011 18012 18013 18014 18015 18016 18017
## [3409] 18018 18019 18020 18021 18022 18023 18024 18025 18026 18027 18028 18029
## [3421] 18030 18031 18032 18033 18034 18035 18036 18037 18038 18039 18040 18041
## [3433] 18042 18043 18044 18045 18046 18047 18048 18049 18050 18051 18052 18053
## [3445] 18054 18055 18056 18057 18058 18059 18060 18061 18062 18063 18064 18065
## [3457] 18066 18067 18068 18069 18070 18071 18072 18073 18074 18075 18076 18077
## [3469] 18078 18079 18080 18081 18082 18083 18084 18085 18086 18087 18088 18089
## [3481] 18090 18091 18092 18093 18094 18095 18096 18097 18098 18099 18100 18101
## [3493] 18102 18103 18104 18105 18106 18107 18108 18109 18110 18111 18112 18113
## [3505] 18114 18115 18116 18117 18118 18119 18120 18121 18122 18123 18124 18125
## [3517] 18126 18127 18128 18129 18130 18131 18132 18133 18134 18135 18136 18137
## [3529] 18138 18139 18140 18141 18142 18143 18144 18145 18146 18147 18148 18149
```

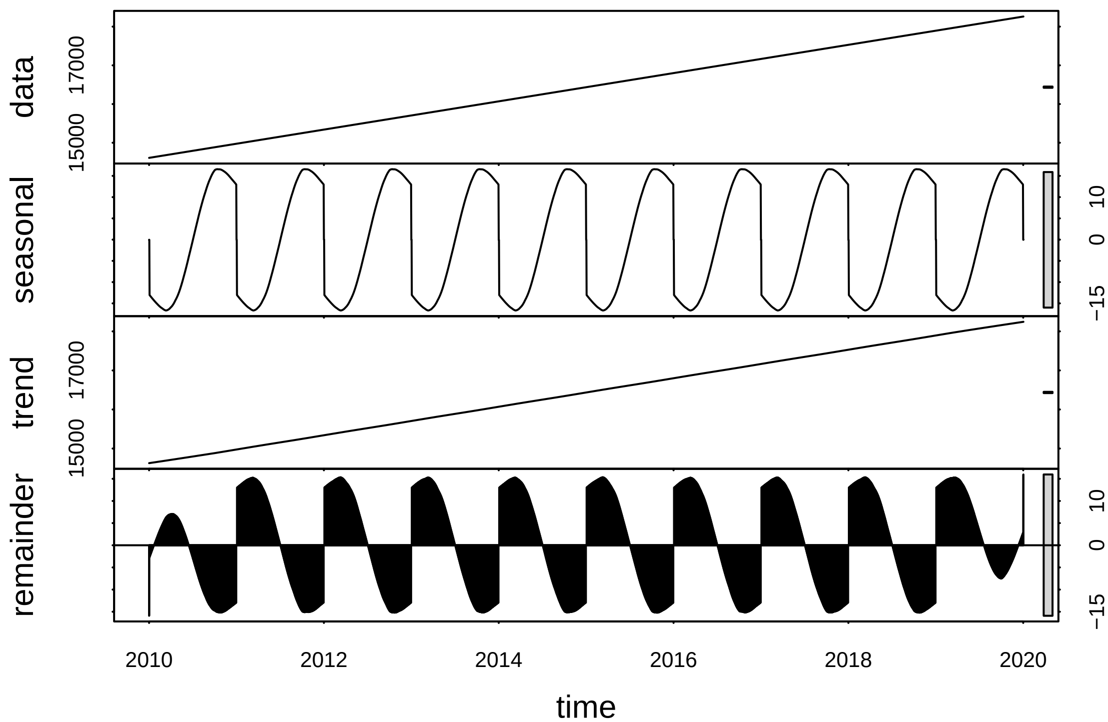
```
## [3541] 18150 18151 18152 18153 18154 18155 18156 18157 18158 18159 18160 18161
## [3553] 18162 18163 18164 18165 18166 18167 18168 18169 18170 18171 18172 18173
## [3565] 18174 18175 18176 18177 18178 18179 18180 18181 18182 18183 18184 18185
## [3577] 18186 18187 18188 18189 18190 18191 18192 18193 18194 18195 18196 18197
## [3589] 18198 18199 18200 18201 18202 18203 18204 18205 18206 18207 18208 18209
## [3601] 18210 18211 18212 18213 18214 18215 18216 18217 18218 18219 18220 18221
## [3613] 18222 18223 18224 18225 18226 18227 18228 18229 18230 18231 18232 18233
## [3625] 18234 18235 18236 18237 18238 18239 18240 18241 18242 18243 18244 18245
## [3637] 18246 18247 18248 18249 18250 18251 18252 18253 18254 18255 18256 18257
## [3649] 18258 18259 18260 18261
```

```r
#monthly set
f_month <- month(first(GaringerOzone_monthly$Date))
f_year <- year(first(GaringerOzone_monthly$Date))

GaringerOzone.monthly.ts <- ts(GaringerOzone_monthly$Date, start = c(f_year, f_month), frequency=12)
print(GaringerOzone.monthly.ts)
```

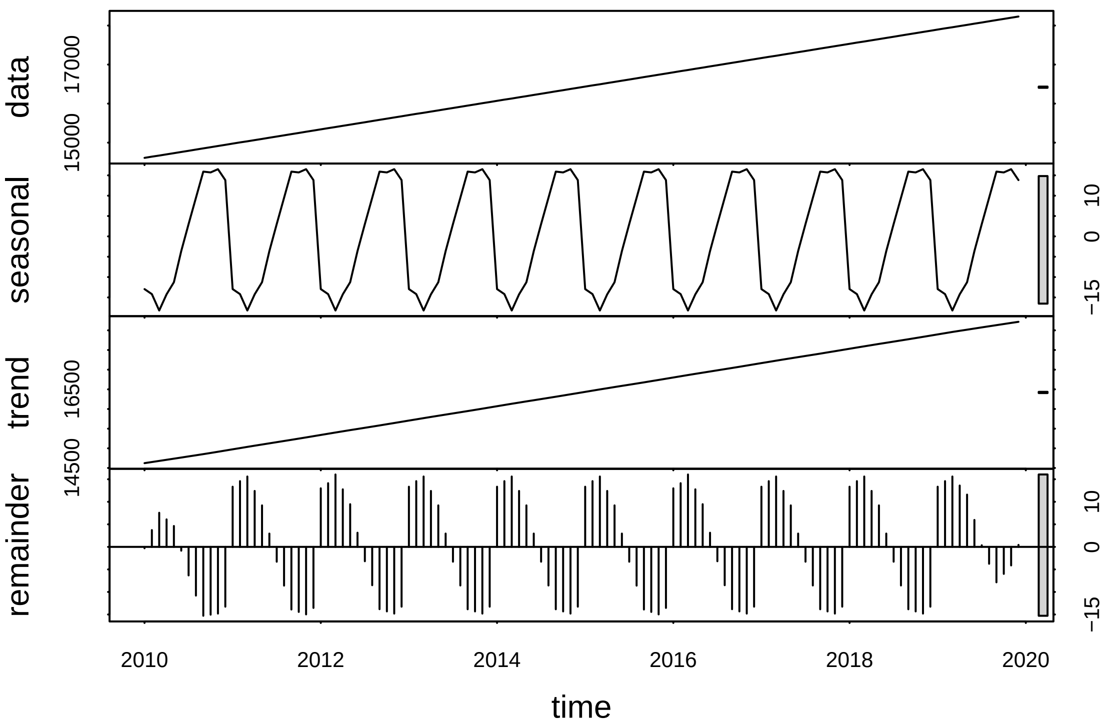
```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 2010 14610 14641 14669 14700 14730 14761 14791 14822 14853 14883 14914 14944
## 2011 14975 15006 15034 15065 15095 15126 15156 15187 15218 15248 15279 15309
## 2012 15340 15371 15400 15431 15461 15492 15522 15553 15584 15614 15645 15675
## 2013 15706 15737 15765 15796 15826 15857 15887 15918 15949 15979 16010 16040
## 2014 16071 16102 16130 16161 16191 16222 16252 16283 16314 16344 16375 16405
## 2015 16436 16467 16495 16526 16556 16587 16617 16648 16679 16709 16740 16770
## 2016 16801 16832 16861 16892 16922 16953 16983 17014 17045 17075 17106 17136
## 2017 17167 17198 17226 17257 17287 17318 17348 17379 17410 17440 17471 17501
## 2018 17532 17563 17591 17622 17652 17683 17713 17744 17775 17805 17836 17866
## 2019 17897 17928 17956 17987 18017 18048 18078 18109 18140 18170 18201 18231
```

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

```r
#11

#generate decomposition of data
G_O3_daily_decomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(G_O3_daily_decomp)
```




```r
#visualize the decomposed series
G_O3_monthly_decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(G_O3_monthly_decomp)
```



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

```r
#12

#run test
G_O3_monthly_SMK1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
#inspect results
G_O3_monthly_SMK1
```

```
## tau = 1, 2-sided pvalue =< 2.22e-16
```

```r
summary(G_O3_monthly_SMK1)
```

```
## Score =  540 , Var(Score) = 1500
## denominator =  540
## tau = 1, 2-sided pvalue =< 2.22e-16
```

```r
#run 2nd test
G_O3_monthly_SMK2 <- trend::smk.test(GaringerOzone.monthly.ts)
#inspect results
G_O3_monthly_SMK2
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = 13.917, p-value < 2.2e-16
```

```
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  540 1500
```

```
summary(G_O3_monthly_SMK2)
```

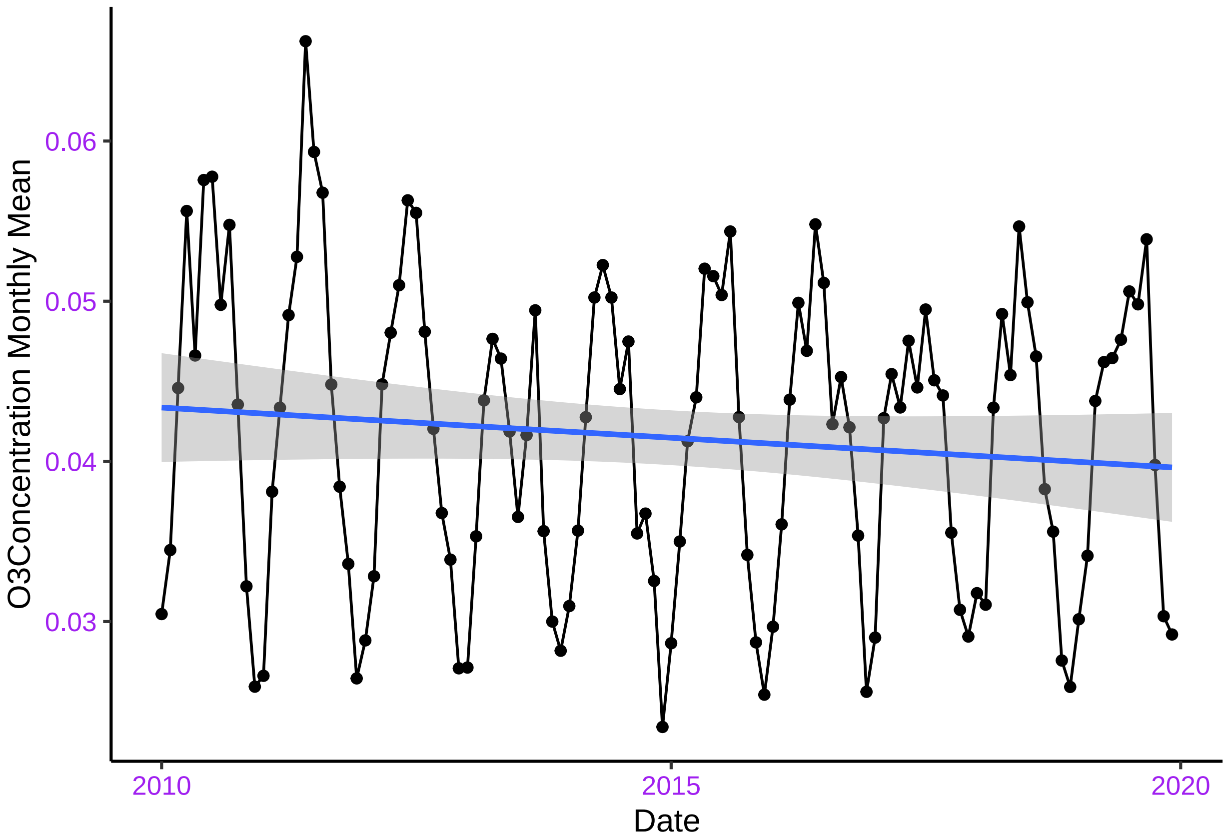
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                      S varS tau     z   Pr(>|z|)
## Season 1:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 2:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 3:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 4:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 5:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 6:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 7:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 8:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 9:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 10:   S = 0 45  125   1 3.935 8.3031e-05 ***
## Season 11:   S = 0 45  125   1 3.935 8.3031e-05 ***
## Season 12:   S = 0 45  125   1 3.935 8.3031e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: The seasonal trend shows up in the monthly data and the only monotonic trend analysis method that can handle seasonality is the Mann-Kendall test. The other method options can be used, but only if the seasonality component is removed ahead of 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

G_O3_m_plot <-
ggplot(GaringerOzone_monthly, aes(x = Date, y = Mean_Ozone)) +
  geom_point() +
  geom_line() +
  ylab("O3Concentration Monthly Mean") +
  geom_smooth( method = lm )
print(G_O3_m_plot)
```

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



14. To accompany your graph, summarize your results in context of the research question (Have ozone concentrations changed over the 2010s at this station?).

Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

> Answer: Decomposing the data confirmed the presence of a seasonal trend in the data. Then the Seasonal Mann Kendall test was used to test stationarity for monotonic trends. For each season of the year represented by tau, we had a value of 1, and a p-vlaue smaller than 0.05 meaning we reject the null hypothesis and have a trend.

> The SMK.Test was used, we saw statistical levels of pronounced results of tau for each season of the year represented by S in the first column. We also see p-values for each season as well, which represent the presence of a change in the trend. The seasonal S-values showed a consistent, positive trend meaning that was moderately pronounced, and an overall increasing trend over time as shown by the positive S-values. The seasonal p-values were also small overall, less than 0.05, demonstrating a non-stationary change in the seasonal trend over time.

> To answer the research question, yes the ozone concentrations have increased over the 2010s at this station.

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

#Separate the components and turn them into data frame
```

```r
GaringerOzone.monthly_Components <- as.data.frame(G_O3_monthly_decomp$time.series[,1:3])
#Exclude seasonal column and make new data frame
GaringerO3.monthly_NOseasonal <- GaringerOzone.monthly_Components %>%
  select(trend, remainder) #made df without seasonal column

#make new time series with no seasonality component
#create second monthly set with old monthly df that has the date info
f_month2 <- month(first(GaringerOzone_monthly$Date))
f_year2 <- year(first(GaringerOzone_monthly$Date))
#make 2nd ts (no seasonality) for tests
GaringerOzone.monthly.ts2 <- ts(GaringerOzone_monthly$Date, start = c(f_year2, f_month2), frequency=12)
print(GaringerOzone.monthly.ts2)
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 2010 14610 14641 14669 14700 14730 14761 14791 14822 14853 14883 14914 14944
## 2011 14975 15006 15034 15065 15095 15126 15156 15187 15218 15248 15279 15309
## 2012 15340 15371 15400 15431 15461 15492 15522 15553 15584 15614 15645 15675
## 2013 15706 15737 15765 15796 15826 15857 15887 15918 15949 15979 16010 16040
## 2014 16071 16102 16130 16161 16191 16222 16252 16283 16314 16344 16375 16405
## 2015 16436 16467 16495 16526 16556 16587 16617 16648 16679 16709 16740 16770
## 2016 16801 16832 16861 16892 16922 16953 16983 17014 17045 17075 17106 17136
## 2017 17167 17198 17226 17257 17287 17318 17348 17379 17410 17440 17471 17501
## 2018 17532 17563 17591 17622 17652 17683 17713 17744 17775 17805 17836 17866
## 2019 17897 17928 17956 17987 18017 18048 18078 18109 18140 18170 18201 18231
```

```r
is.ts(GaringerOzone.monthly.ts2)#check it is a ts and no longer a df
```

```
## [1] TRUE
```

```r
                                # WOOHOO!!! I figured it out! (took a while)

#16

#run test
G_O3_monthly_SMK1.2 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts2)
#inspect results
G_O3_monthly_SMK1.2
```

```
## tau = 1, 2-sided pvalue =< 2.22e-16
```

```r
summary(G_O3_monthly_SMK1.2)
```

```
## Score =  540 , Var(Score) = 1500
## denominator =  540
## tau = 1, 2-sided pvalue =< 2.22e-16
```

```r
#run 2nd test
G_O3_monthly_SMK2.2 <- trend::smk.test(GaringerOzone.monthly.ts2)
#inspect results
G_O3_monthly_SMK2.2
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts2
## z = 13.917, p-value < 2.2e-16
## alternative hypothesis: true S is not equal to 0
```

```
## sample estimates:
##    S varS
##  540 1500
```

```
summary(G_O3_monthly_SMK2.2)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts2
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                      S varS tau     z   Pr(>|z|)
## Season 1:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 2:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 3:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 4:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 5:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 6:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 7:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 8:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 9:   S = 0  45  125   1 3.935 8.3031e-05 ***
## Season 10:   S = 0 45  125   1 3.935 8.3031e-05 ***
## Season 11:   S = 0 45  125   1 3.935 8.3031e-05 ***
## Season 12:   S = 0 45  125   1 3.935 8.3031e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: After removing the seasonal component, and running the Seasonal Mann Kendall test, tau stayed at a value of 1, and the p-value again was less than 0.05 giving support to reject the null hypothesis. This means we do have the presence of a trend still.

When the SMK.Test was used, the seasonal S-values showed a consistent, positive trend STILL meaning that was moderately pronounced, and an overall increasing trend over time as shown by the positive S-values. The seasonal p-values were also STILL small overall, less than 0.05, demonstrating a non-stationary change in the seasonal trend over time.

Removing the seasonality component did not seem to change the results in the SeasonalMannKendall and smk.test tests.