Mill Creek Water Analysis

John C. Ayers

This notebook investigates four water quality measures that have been measured by the USGS since 2007: temperature, conductivity (a measure of salinity), pH, and dissolved oxygen. The site is USGS 03431083 MILL CREEK AT LEBANON ROAD AT NASHVILLE, TN. Information on the site is at https://waterdata.usgs.gov/nwis/inventory/?site_no=03431083&agency_cd=USGS (https://waterdata.usgs.gov/nwis/inventory/?site_no=03431083&agency_cd=USGS) Location: Latitude 36°09’50“, Longitude 86°41’57” NAD27

Useful packages

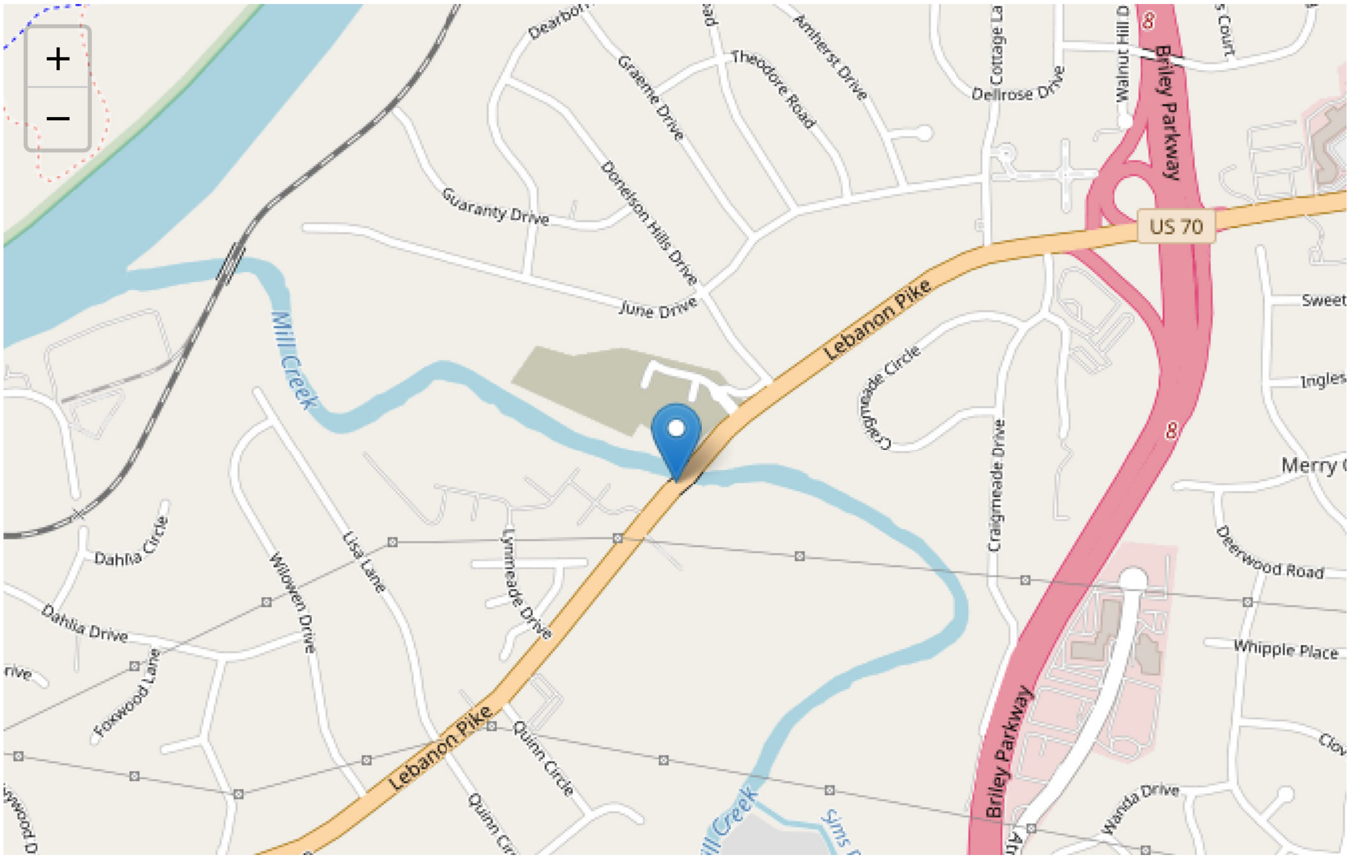
```
if (!require("pacman"))
   install.packages("pacman")
```

```
## Loading required package: pacman
```

```
pacman::p_load(tidyverse, lubridate, skimr, timetk, leaflet, psych)
```

Map of site location

```
m <- leaflet(df) %>% addMarkers(lat = 36.16389, lng = -86.69917) %>% addTiles()
m
```



Leaflet (https://leafletjs.com) | © OpenStreetMap (http://openstreetmap.org) contributors, CC-BY-SA (http://creativecommons.org/licenses/by-sa/2.0/)

Import and clean data

```
# Unfortunately this USGS station does not have discharge data
# The begin and end dates can be changed in the url below
Data <- read_delim("https://nwis.waterdata.usgs.gov/usa/nwis/uv/?cb_00095=on&cb_00300=on&cb_0001
0=on&cb_00400=on&format=rdb&site_no=03431083&period=&begin_date=2019-12-22&end_date=2020-12-21",
"\t", escape_double = FALSE, col_names = TRUE, locale = locale(tz = "America/Chicago"), trim_ws
 = TRUE, skip = 30)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   agency_cd = col_character(),
##   site_no = col_character(),
##   datetime = col_character(),
##   tz_cd = col_character(),
##   `131093_00010` = col_character(),
##   `131093_00010_cd` = col_character(),
##   `131094_00095` = col_character(),
##   `131094_00095_cd` = col_character(),
##   `131095_00400` = col_character(),
##   `131095_00400_cd` = col_character(),
##   `131096_00300` = col_character(),
##   `131096_00300_cd` = col_character()
## )
```

```
Data <- Data[-1,-(1:2)]
Data <- rename(Data, Temp_C = `131093_00010`, SpC = `131094_00095`, pH = `131095_00400`, DO = `1
31096_00300`)
Data <- select(Data, datetime, Temp_C, SpC, pH, DO)
Data <- type_convert(Data)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   datetime = col_datetime(format = ""),
##   Temp_C = col_double(),
##   SpC = col_double(),
##   pH = col_double(),
##   DO = col_double()
## )
```

```
skim(Data)
```

Data summary

Name	Data
Number of rows	17555
Number of columns	5
_______________________	
Column type frequency:	
numeric	4
POSIXct	1
________________________	
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Temp_C	421	0.98	17.12	6.24	5.0	11.60	16.3	23.2	30.5	▅▇▆▆▃
SpC	242	0.99	482.85	78.21	121.0	449.00	494.0	543.0	623.0	▁▁▂▇▆
pH	185	0.99	7.99	0.21	7.2	7.90	8.0	8.1	8.8	▁▂▇▂▁
DO	273	0.98	8.82	2.24	2.8	7.23	8.9	10.4	16.6	▂▇▇▃▁

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
datetime	0	1	2019-12-22	2020-12-21 23:30:00	2020-06-22 01:00:00	17553

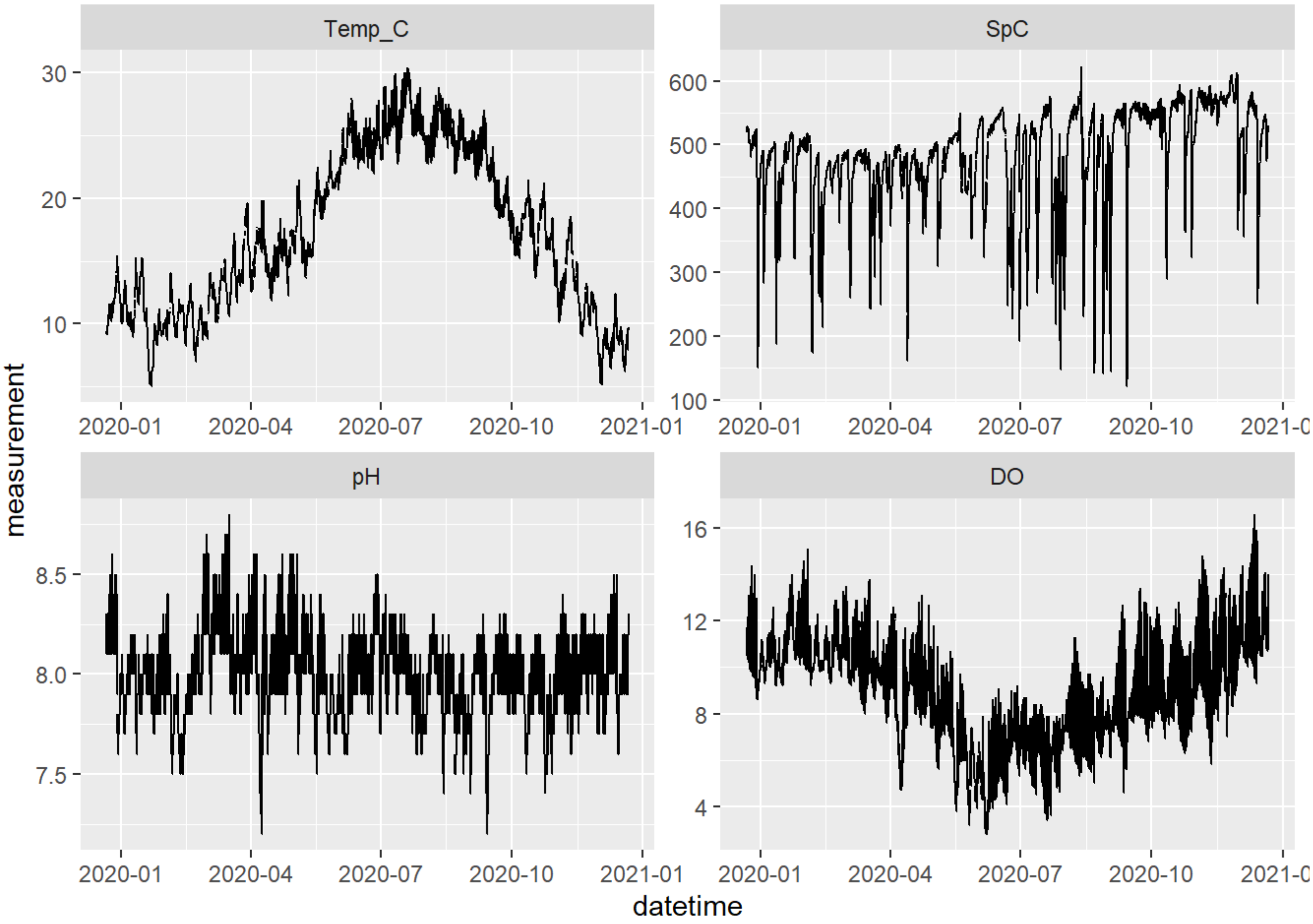
```
write_excel_csv(Data, "MillCreekDataWQ.csv", col_names = TRUE)
```

Plots of temporal trends

```
DataLong <-  gather(Data, variable, measurement, Temp_C:DO, factor_key=TRUE)
MinDate <- as.Date(min(Data$datetime))
MaxDate <- as.Date(max(Data$datetime))
print(paste("From ", MinDate, " to ", MaxDate))
```

```
## [1] "From  2019-12-22  to  2020-12-21"
```

```
print(ggplot(DataLong, aes(x = datetime, y = measurement)) +
  geom_line() +
  scale_x_datetime(date_labels = "%Y-%m") +
  facet_wrap(~ variable, scales = "free", nrow = 2))
```



Create seasons

```
Data <- Data %>%
  add_column(Season = NA)
for (i in 1:nrow(Data)) {
  if (as.Date(Data$datetime[i]) >= as.Date("2019-12-21") & as.Date(Data$datetime[i]) <= as.Date(
"2020-03-19")) {
  Data$Season[i] = paste("Winter")
  }
  if (as.Date(Data$datetime[i]) >= as.Date("2020-03-20") & as.Date(Data$datetime[i]) <= as.Date(
"2020-06-19")) {
  Data$Season[i] = paste("Spring")
  }
   if (as.Date(Data$datetime[i]) >= as.Date("2020-06-20") & as.Date(Data$datetime[i]) <= as.Date
("2020-09-21")) {
  Data$Season[i] = paste("Summer")
  }
   if (as.Date(Data$datetime[i]) >= as.Date("2020-09-22") & as.Date(Data$datetime[i]) <= as.Date
("2020-12-20")) {
  Data$Season[i] = paste("Fall")
   }
}
```

Summary statistics by season

```
Data$Season <- factor(Data$Season)
textvars <- c("Temp_C", "SpC", "pH", "DO")
describeBy(Data[textvars], list(Data$Season))
```

```
## 
##  Descriptive statistics by group 
## : Fall
##        vars    n   mean    sd median trimmed   mad   min   max range  skew
## Temp_C    1 4228  13.95  4.28   14.3   14.04  5.63   5.2  21.8  16.6 -0.15
## SpC       2 4274 539.85 54.47  557.0  550.73 23.72 251.0 613.0 362.0 -2.26
## pH        3 4285   7.97  0.18    8.0    7.98  0.15   7.4   8.5   1.1 -0.17
## DO        4 4274   9.96  1.90    9.9    9.86  2.08   5.8  16.6  10.8  0.43
##        kurtosis   se
## Temp_C    -1.25 0.07
## SpC        5.68 0.83
## pH         0.14 0.00
## DO        -0.26 0.03
## ------------------------------------------------------------ 
## : Spring
##        vars    n   mean    sd median trimmed   mad   min   max range  skew
## Temp_C    1 4212  18.24  3.84  17.05   18.00  4.15  11.7  28.0  16.3  0.46
## SpC       2 4285 468.62 53.88 476.00  473.10 47.44 162.0 559.0 397.0 -1.18
## pH        3 4347   8.00  0.22   8.00    8.00  0.15   7.2   8.6   1.4 -0.07
## DO        4 4313   7.58  2.10   7.70    7.59  2.37   2.8  13.1  10.3 -0.07
##        kurtosis   se
## Temp_C    -0.91 0.06
## SpC        2.99 0.82
## pH         0.72 0.00
## DO        -0.70 0.03
## ------------------------------------------------------------ 
## : Summer
##        vars    n   mean    sd median trimmed   mad   min   max range  skew
## Temp_C    1 4426  25.07  2.06   25.0   25.05  1.93  19.0  30.5  11.5  0.02
## SpC       2 4477 471.42 97.02  513.0  486.02 63.75 121.0 623.0 502.0 -1.20
## pH        3 4467   7.96  0.20    8.0    7.97  0.15   7.2   8.5   1.3 -0.43
## DO        4 4441   7.17  1.38    7.2    7.13  1.19   3.4  12.7   9.3  0.45
##        kurtosis   se
## Temp_C     0.11 0.03
## SpC        0.67 1.45
## pH         0.46 0.00
## DO         1.17 0.02
## ------------------------------------------------------------ 
## : Winter
##        vars    n   mean    sd median trimmed   mad   min   max range  skew
## Temp_C    1 4220  10.94  1.98   10.9   10.97  1.93   5.0  16.3  11.3 -0.21
## SpC       2 4229 451.32 67.09  476.0  463.90 35.58 151.0 528.0 377.0 -1.84
## pH        3 4223   8.03  0.24    8.0    8.02  0.15   7.5   8.8   1.3  0.39
## DO        4 4206  10.63  1.11   10.4   10.53  0.89   8.0  15.1   7.1  0.88
##        kurtosis   se
## Temp_C     0.33 0.03
## SpC        3.29 1.03
## pH         0.11 0.00
## DO         0.70 0.02
```

Average % change by season

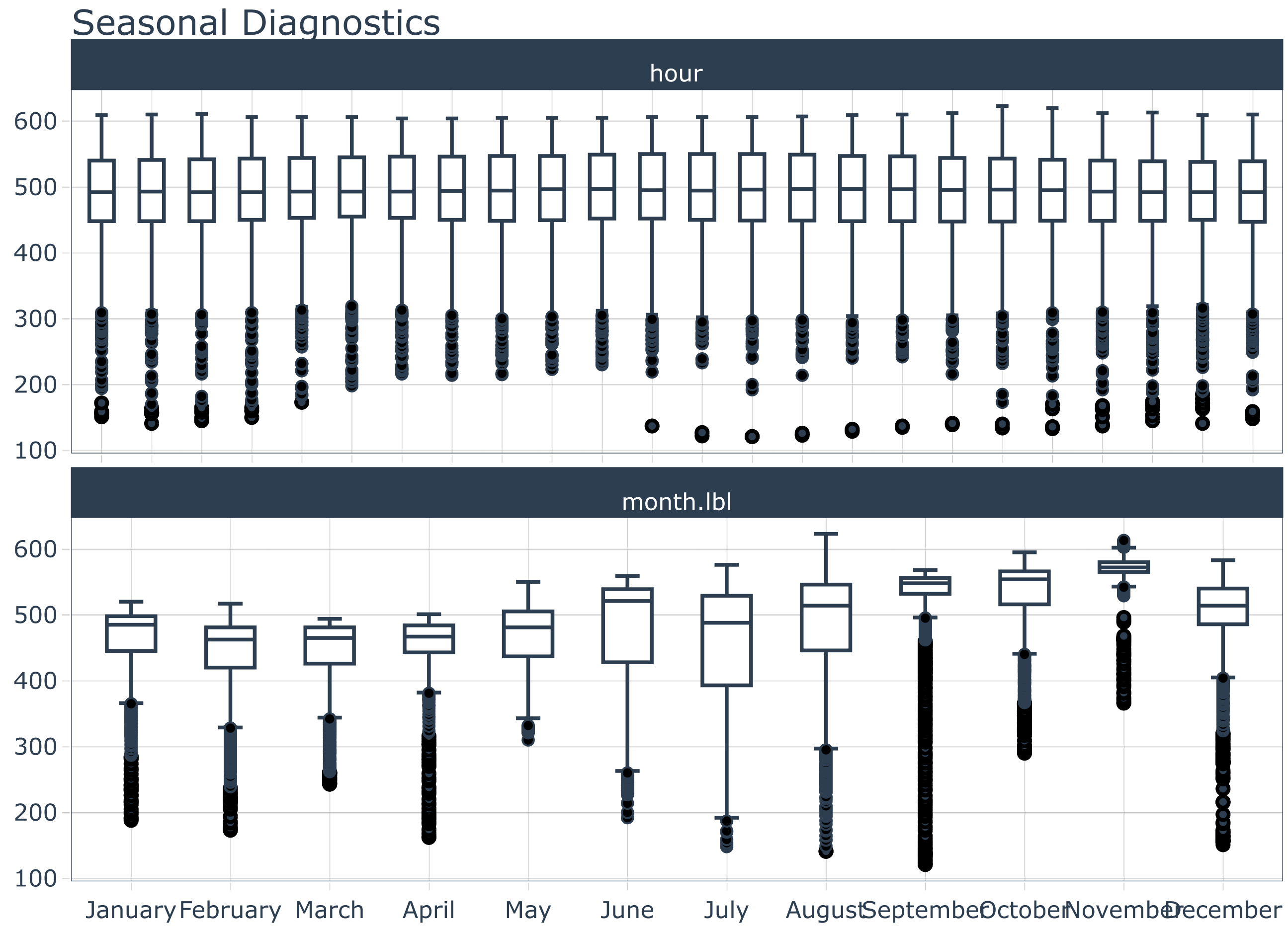
```r
percent_change <- function(x) {
  x <- 100*((x - mean(x, na.rm = TRUE))/mean(x, na.rm = TRUE))
}
season_change <- function(y) {transmute(y, datetime, Season,
                  Temp_C = percent_change(Temp_C),
                  SpC = percent_change(SpC),
                  pH = percent_change(pH),
                  DO = percent_change(DO)
                  )}
Winter <- filter(Data, Season == "Winter")
WinterChange <- season_change(Winter)
Spring <- filter(Data, Season == "Spring")
SpringChange <- season_change(Spring)
Summer <- filter(Data, Season == "Summer")
SummerChange <- season_change(Summer)
Fall <- filter(Data, Season == "Fall")
FallChange <- season_change(Fall)
SeasonChange <- bind_rows(WinterChange, SpringChange, SummerChange, FallChange)
describeBy(SeasonChange[textvars], list(SeasonChange$Season))
```

```
##
##  Descriptive statistics by group
## : Fall
##        vars    n mean    sd median trimmed   mad    min   max  range  skew
## Temp_C    1 4228    0 30.66   2.53    0.64 40.40 -62.71 56.31 119.03 -0.15
## SpC       2 4274    0 10.09   3.18    2.01  4.39 -53.51 13.55  67.06 -2.26
## pH        3 4285    0  2.31   0.32    0.07  1.86  -7.20  6.59  13.79 -0.17
## DO        4 4274    0 19.04  -0.59   -1.03 20.84 -41.76 66.69 108.45  0.43
##        kurtosis   se
## Temp_C    -1.25 0.47
## SpC        5.68 0.15
## pH         0.14 0.04
## DO        -0.26 0.29
## ------------------------------------------------------------
## : Spring
##        vars    n mean    sd median trimmed   mad    min   max  range  skew
## Temp_C    1 4212    0 21.06  -6.51   -1.27 22.76 -35.84 53.54  89.38  0.46
## SpC       2 4285    0 11.50   1.57    0.96 10.12 -65.43 19.29  84.72 -1.18
## pH        3 4347    0  2.73   0.02   -0.01  1.85  -9.98  7.53  17.50 -0.07
## DO        4 4313    0 27.76   1.56    0.11 31.29 -63.07 72.78 135.85 -0.07
##        kurtosis   se
## Temp_C    -0.91 0.32
## SpC        2.99 0.18
## pH         0.72 0.04
## DO        -0.70 0.42
## ------------------------------------------------------------
## : Summer
##        vars    n mean    sd median trimmed   mad    min   max  range  skew
## Temp_C    1 4426    0  8.22  -0.28   -0.08  7.69 -24.22 21.65  45.87  0.02
## SpC       2 4477    0 20.58   8.82    3.10 13.52 -74.33 32.15 106.49 -1.20
## pH        3 4467    0  2.54   0.51    0.10  1.86  -9.54  6.79  16.33 -0.43
## DO        4 4441    0 19.23   0.35   -0.62 16.53 -52.61 77.01 129.62  0.45
##        kurtosis   se
## Temp_C     0.11 0.12
## SpC        0.67 0.31
## pH         0.46 0.04
## DO         1.17 0.29
## ------------------------------------------------------------
## : Winter
##        vars    n mean    sd median trimmed   mad    min   max  range  skew
## Temp_C    1 4220    0 18.07  -0.35    0.25 17.62 -54.29 49.02 103.31 -0.21
## SpC       2 4229    0 14.87   5.47    2.79  7.88 -66.54 16.99  83.53 -1.84
## pH        3 4223    0  2.94  -0.36   -0.12  1.85  -6.58  9.61  16.19  0.39
## DO        4 4206    0 10.41  -2.17   -0.97  8.37 -24.74 42.04  66.79  0.88
##        kurtosis   se
## Temp_C     0.33 0.28
## SpC        3.29 0.23
## pH         0.11 0.05
## DO         0.70 0.16
```

Conductivity seasonal diagnostics

```
plot_seasonal_diagnostics(Data, datetime, SpC, .feature_set = c("hour", "month.lbl"), .interacti
ve = TRUE)
```

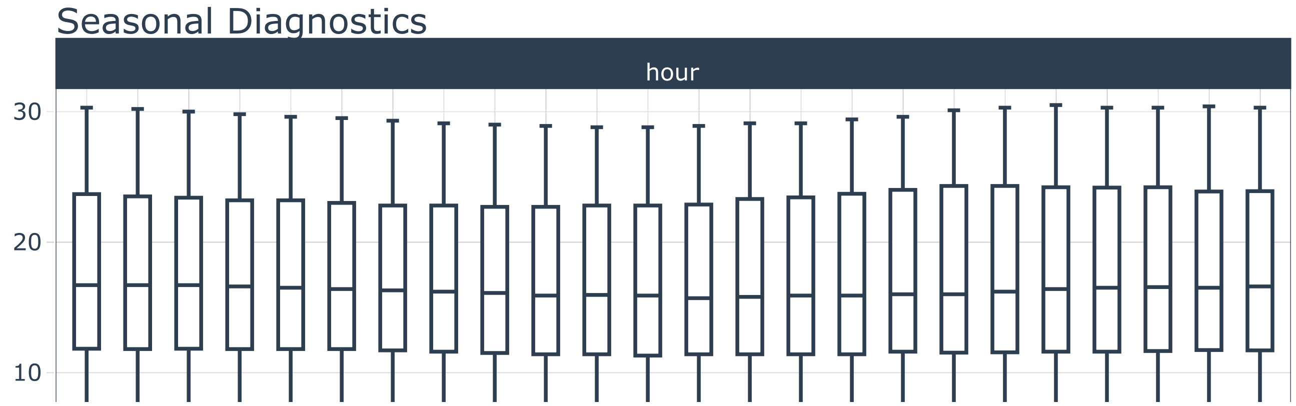
```
## Warning: Removed 484 rows containing non-finite values (stat_boxplot).
```

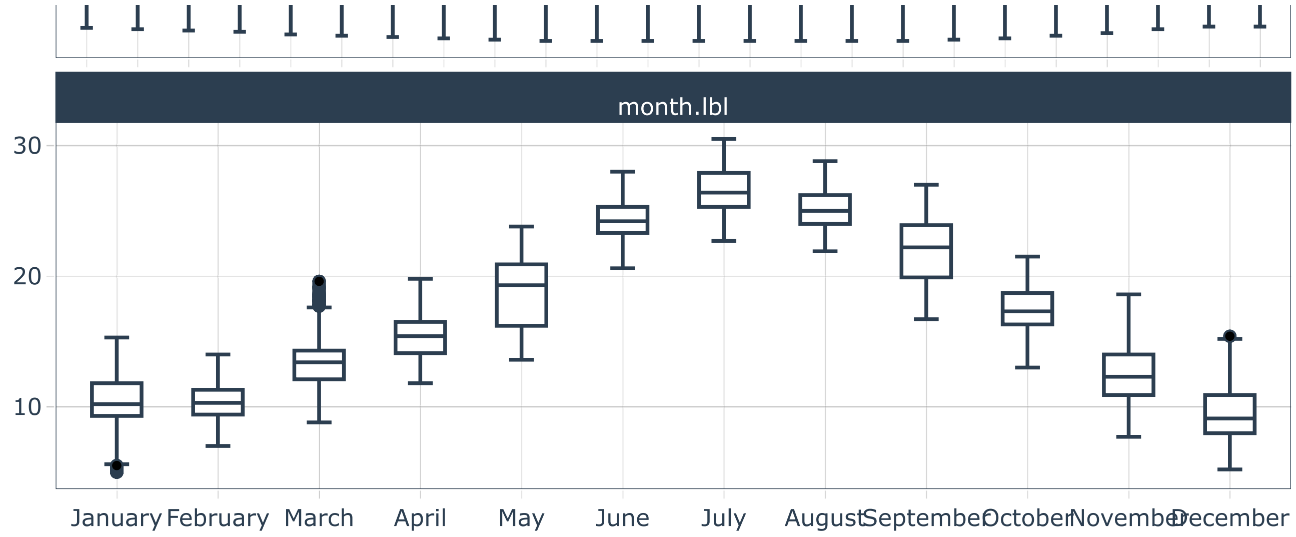


Temp seasonal diagnostics

```
plot_seasonal_diagnostics(Data, datetime, Temp_C, .feature_set = c("hour", "month.lbl"), .intera
ctive = TRUE)
```

```
## Warning: Removed 842 rows containing non-finite values (stat_boxplot).
```

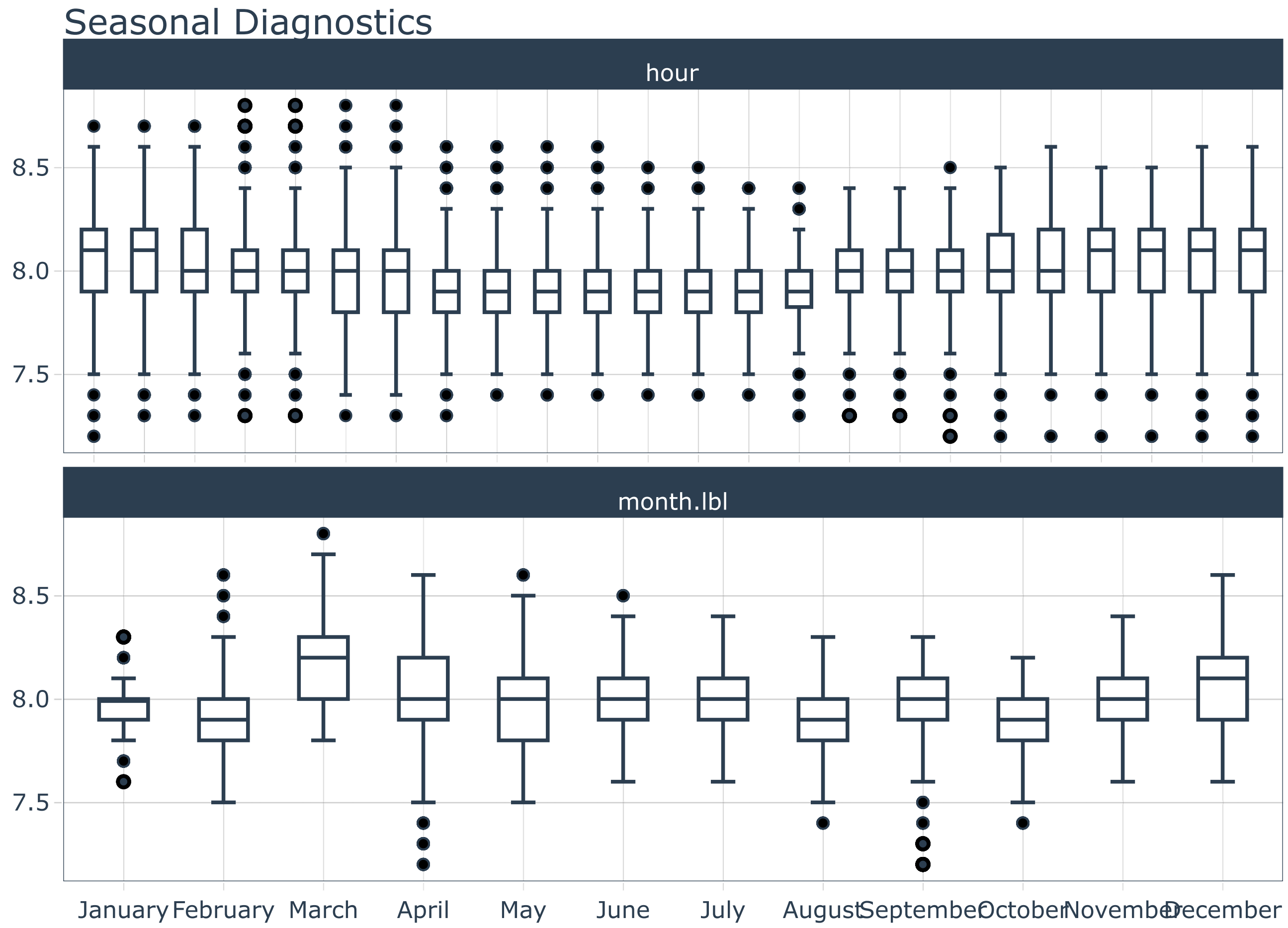




pH seasonal diagnostics

```
plot_seasonal_diagnostics(Data, datetime, pH, .feature_set = c("hour", "month.lbl"), .interactiv
e = TRUE)
```

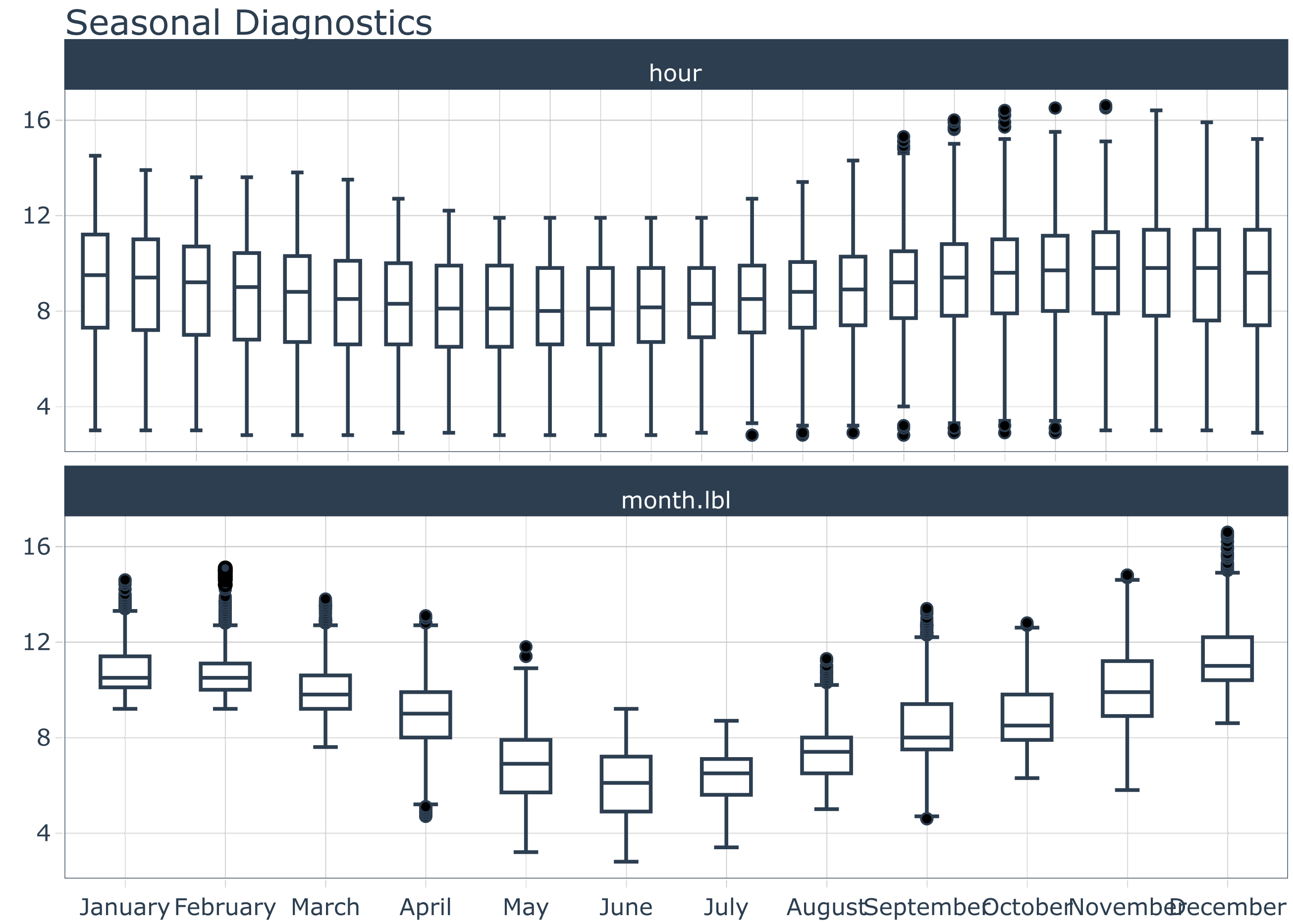
```
## Warning: Removed 370 rows containing non-finite values (stat_boxplot).
```



DO seasonal diagnostics

```
plot_seasonal_diagnostics(Data, datetime, DO, .feature_set = c("hour", "month.lbl"), .interactiv
e = TRUE)
```

```
## Warning: Removed 546 rows containing non-finite values (stat_boxplot).
```



```
sessionInfo()
```

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18363)
## 
## Matrix products: default
## 
## locale:
## [1] LC_COLLATE=English_United States.1252 
## [2] LC_CTYPE=English_United States.1252   
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C                          
## [5] LC_TIME=English_United States.1252    
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
##  [1] psych_2.0.12      leaflet_2.0.4.1   timetk_2.6.1      skimr_2.1.3      
##  [5] lubridate_1.7.9.2 forcats_0.5.1     stringr_1.4.0     dplyr_1.0.4      
##  [9] purrr_0.3.4       readr_1.4.0       tidyr_1.1.2       tibble_3.0.6     
## [13] ggplot2_3.3.3     tidyverse_1.3.0   pacman_0.5.1     
## 
## loaded via a namespace (and not attached):
##  [1] colorspace_2.0-0   ellipsis_0.3.1     class_7.3-17       base64enc_0.1-3   
##  [5] fs_1.5.0           rstudioapi_0.13    farver_2.0.3       listenv_0.8.0     
##  [9] furrr_0.2.2        dials_0.0.9        prodlim_2019.11.13 xml2_1.3.2        
## [13] codetools_0.2-16   splines_4.0.3      mnormt_2.0.2       knitr_1.31        
## [17] jsonlite_1.7.2     workflows_0.2.1    pROC_1.17.0.1      broom_0.7.4       
## [21] dbplyr_2.1.0       yardstick_0.0.7    tune_0.1.2         compiler_4.0.3    
## [25] httr_1.4.2         backports_1.2.1    lazyeval_0.2.2     assertthat_0.2.1  
## [29] Matrix_1.2-18      cli_2.3.0          htmltools_0.5.1.1  tools_4.0.3       
## [33] gtable_0.3.0       glue_1.4.2         Rcpp_1.0.6         cellranger_1.1.0  
## [37] DiceDesign_1.8-1   vctrs_0.3.7        nlme_3.1-149       iterators_1.0.13  
## [41] crosstalk_1.1.1    parsnip_0.1.5      timeDate_3043.102  gower_0.2.2       
## [45] xfun_0.20          globals_0.14.0     rvest_0.3.6        lifecycle_0.2.0   
## [49] future_1.21.0      MASS_7.3-53        zoo_1.8-8          scales_1.1.1      
## [53] ipred_0.9-9        hms_1.0.0          parallel_4.0.3     curl_4.3          
## [57] yaml_2.2.1         rpart_4.1-15       stringi_1.5.3      highr_0.8         
## [61] foreach_1.5.1      lhs_1.1.1          lava_1.6.8.1       repr_1.1.3        
## [65] rlang_0.4.10       pkgconfig_2.0.3    rsample_0.0.8      evaluate_0.14     
## [69] lattice_0.20-41    labeling_0.4.2     recipes_0.1.15     htmlwidgets_1.5.3 
## [73] tidyselect_1.1.0   parallelly_1.23.0  plyr_1.8.6         magrittr_2.0.1    
## [77] R6_2.5.0           generics_0.1.0     DBI_1.1.1          pillar_1.4.7      
## [81] haven_2.3.1        withr_2.4.1        xts_0.12.1         survival_3.2-7    
## [85] nnet_7.3-14        modelr_0.1.8       crayon_1.4.0       plotly_4.9.3      
## [89] tmvnsim_1.0-2      rmarkdown_2.6      grid_4.0.3         readxl_1.3.1      
## [93] data.table_1.13.6  reprex_1.0.0       digest_0.6.27      GPfit_1.0-8       
## [97] munsell_0.5.0      viridisLite_0.3.0 
```

```
save.image()
```