Plotting Change Through Time

Author: Cole Brookson **Date:** 20 July 2022

When choosing to plot anything to do with time, the time is almost always displayed on the x-axis. This makes sense as we typically read plots as we read English language text - left to right. Thus, we will always plot time as increasing from left to right on our plots. By FAR the most common way to plot change through time is to plot it as a line plot, optionally with points added in. Less common in EEB-type data (but still useful) are box plots and bar plots.

Timeseries Plots (Line Plots)

Line plots, also referred to as timeseries plots colloquially, are plots that have the focal variable of interest on the y-axis, and some time variable on the x-axis. To follow our (LINK TO ITERATIVE PLOTTING) iterative plotting method, we'll start one of these plots and build up.

For this, we'll use our example with data on temperature in Alaska from 1988 to present. See the full tutorial here (LINK TO PLOTTING WITH MULTIPLE DATAFRAMES):

Just the Code:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

```
# add in an option to keep dplyr quiet
options(dplyr.summarise.inform = FALSE)
library(lterdatasampler)
library(lubridate)
```

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

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

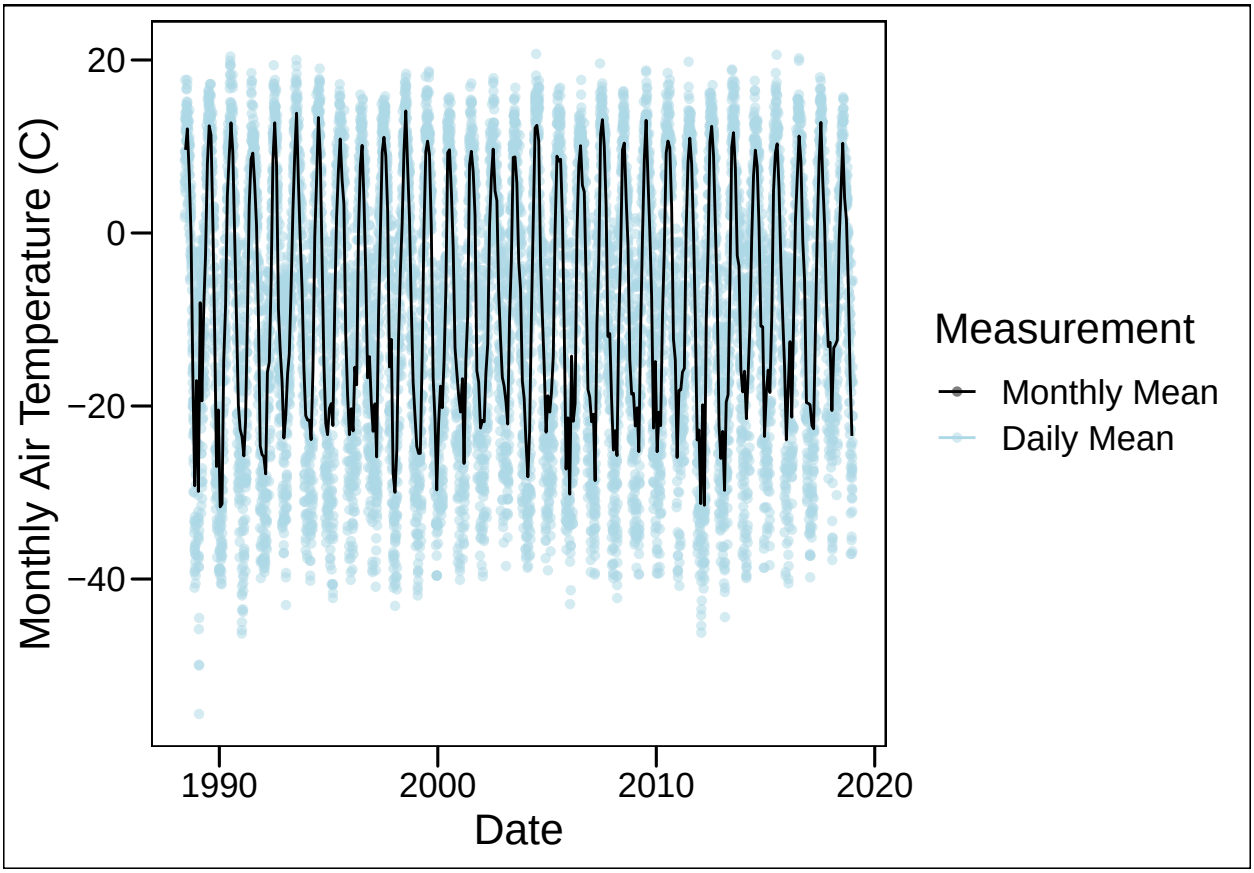
```
library(ggthemes)

df = lterdatasampler::arc_weather

# make sub-groups for the dates
df = df %>%
  dplyr::rowwise() %>%
  dplyr::mutate(
    month = lubridate::month(date),
    year = lubridate::year(date)
  )
# make separate dataframe for monthly measurements
```

```r
df_monthly = df %>%
  dplyr::select(year, month, mean_airtemp) %>%
  dplyr::group_by(year, month) %>%
  dplyr::summarize(
    month_airtemp = mean(mean_airtemp, na.rm = TRUE)
  ) %>%
  # assume all days are the 15th of the month
  dplyr::mutate(
    day = 15
  ) %>%
  # make single date
  dplyr::mutate(
    date = lubridate::make_date(year, month, day)
  )

ggplot() +
  # switched order - daily first
  geom_point(data = df, mapping = aes(x = date, y = mean_airtemp,
                                      colour = "lightblue"), shape = 16,
                                      alpha = 0.5) +
  # monthly second
  geom_line(data = df_monthly, mapping = aes(x = date, y = month_airtemp,
                                             colour = "black")) +
  ggthemes::theme_base() +
  labs(x = "Date", y = "Monthly Air Temperature (C)") +
  guides(
    colour = guide_legend()
  ) +
  # add in colour scale
  scale_colour_identity(guide = "legend",
                        name = "Measurement", # the title of the legend
                        breaks = c("black", "lightblue"), # the colours
                        labels = c("Monthly Mean", "Daily Mean") # legend labels
                        )
```



Box Plot

Box plots are a helpful form of plot to show how data might change through time by summarizing the distribution of said data in some binned form. To use the above example regarding data on temperature, we could use a box plot to compare how the distribution of the daily temperature measurements change through the months.

As a reminder, here's how to interpret the various components of a boxplot:

For this, we'll use our example with data on stream chemistry over time. See the full tutorial here (LINK TO PLOTTING WITH FACETING):

Just the Code

```
library(tidyverse)
library(lterdatasampler)
library(lubridate)

df <- lterdatasampler::luq_streamchem %>%
   # remove columns we don't need
  dplyr::select(sample_date, na, k) %>%
  tidyr::pivot_longer(
    # specify which columns to join together
    cols = c(na, k),
    # specify what the new name of the grouping variable will be
    names_to = "element"
  ) %>%
```

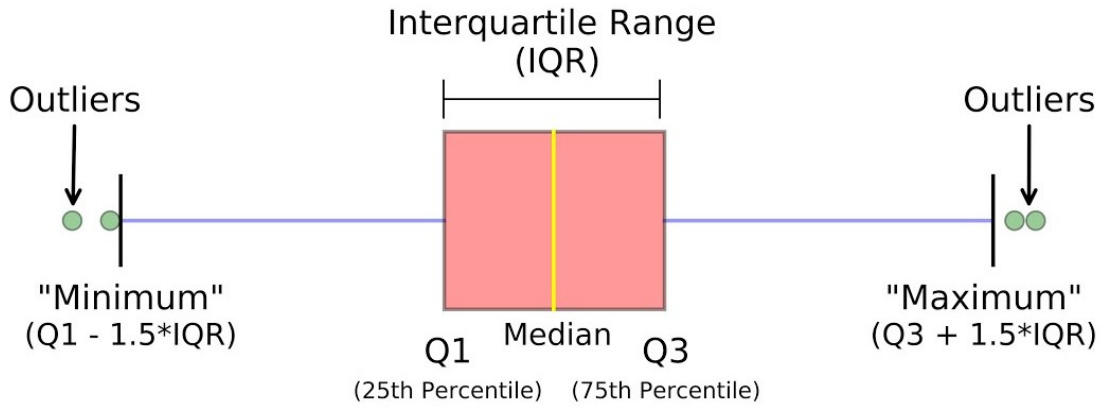
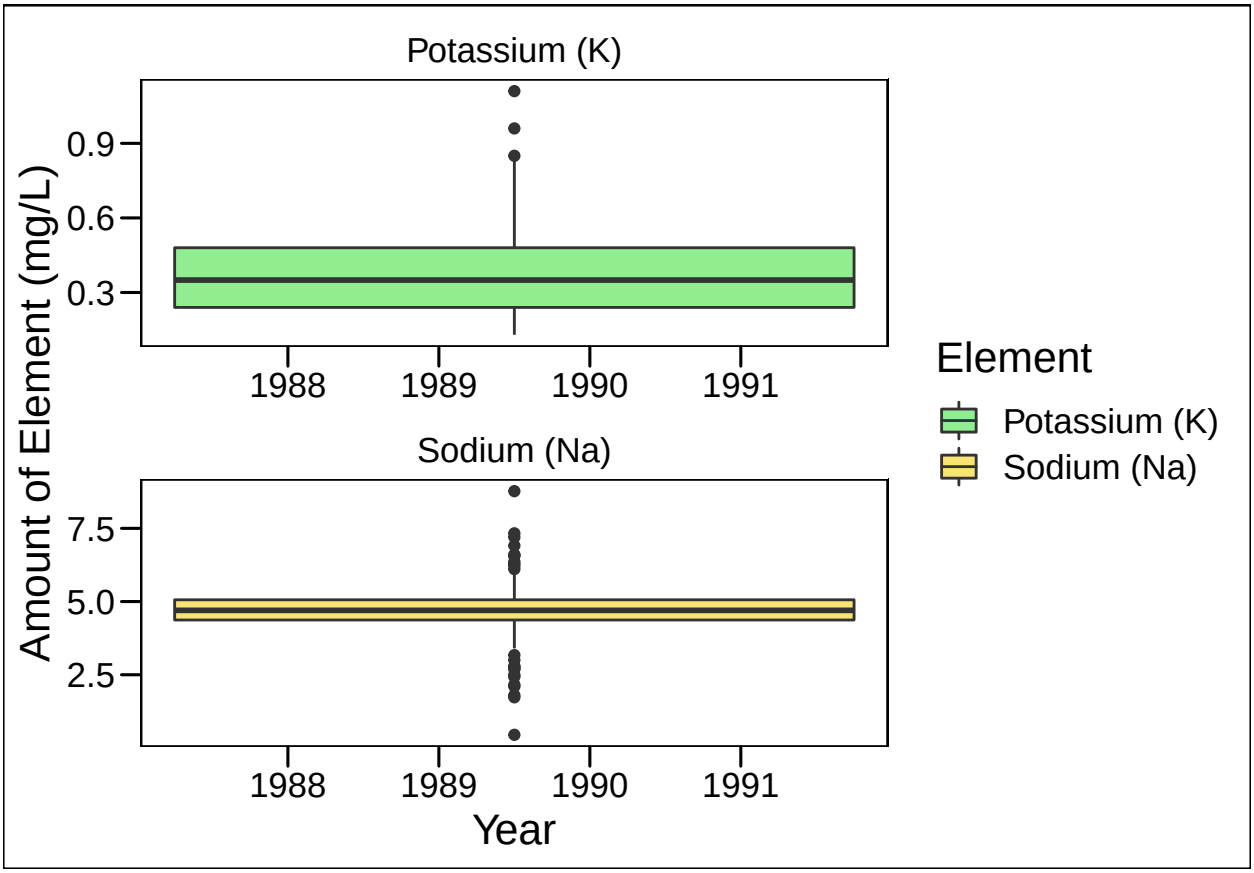


Figure 1: box plot interpretation. credit: Michael Galarnyk

```
  dplyr::mutate(
    year = lubridate::year(sample_date),
    month = lubridate::month(sample_date)
  )

ggplot() +
  geom_boxplot(data = df, mapping = aes(x = year, y = value,
                                        fill = element)) +
  labs(x = "Year", y = "Amount of Element (mg/L)") +
  ggthemes::theme_base() +
  facet_wrap(~element,
             # I want this to be one column with two rows
             ncol = 1,
             scales = "free",
             labeller = as_labeller(c(`k` = "Potassium (K)",
                                      `na` = "Sodium (Na)"))) +
    scale_fill_manual("Element",
                      values = c("#90ee90", "#F8E473"),
                      labels = c("Potassium (K)", "Sodium (Na)"))
```

```
## Warning: Removed 11 rows containing non-finite values (stat_boxplot).
```


Potassium (K)
Sodium (Na)
Amount of Element (mg/L)
0.9
0.6
0.3
7.5
5.0
2.5
1988
1989
1990
1991
Year
Element
Potassium (K)
Sodium (Na)