Climate Session

Kateri Salk

2021-09-20

Intro

For many analyses and model applications, we may want to gather local weather and climate conditions to inform the conditions we observe in aquatic systems. Processing and analyzing NOAA data is an ideal process to automate, since the data downloaded from NOAA's platform have a consistent format.

During this session, we will:

1. Import NOAA data into RStudio two ways
2. Process daily weather data
3. Aggregate precipitation and temperature into monthly and annual datasets
4. Create visualizations of precipitation and temperature data
5. Save processed datasets and graphs

Setup

Acquiring NOAA Weather Data

Manual

1. Navigate to www.ncdc.noaa.gov and choose "Data Access" from the top menu
2. Select "Quick Links"
3. Select "U.S. Local Climatological Data"
4. Select "Local Climatological Data (LCD)"
5. Depending on your location of interest, navigate through the location selector at the top of the page until a weather station of interest appears in the "Station Details" section at the bottom of the page. Select "ADD TO CART" for the station of interest.
6. View your cart from the top of the page
7. Select "LCD CSV" and the appropriate date range (max. of 10 years per order)
8. Submit request and check email
9. Save your file into your raw data folder

```
manitowoc_weather_manual <- read.csv("./Data/Raw/Manitowoc_Weather.csv")
```

R Package: rnoaa

`rnoaa` allows us to download only a year at a time. We will need to download each year individually and then stitch the years together into a dataset. The `lcd` function allows us to download the relevant data.

Documentation can be found here

```
manitowoc_weather_2012 <- lcd(station = "72645594897", year = 2012, verbose = TRUE)
manitowoc_weather_2013 <- lcd(station = "72645594897", year = 2013, verbose = TRUE)
manitowoc_weather_2014 <- lcd(station = "72645594897", year = 2014, verbose = TRUE)
manitowoc_weather_2015 <- lcd(station = "72645594897", year = 2015, verbose = TRUE)
manitowoc_weather_2016 <- lcd(station = "72645594897", year = 2016, verbose = TRUE)
```

```
manitowoc_weather_2017 <- lcd(station = "72645594897", year = 2017, verbose = TRUE)
manitowoc_weather_2018 <- lcd(station = "72645594897", year = 2018, verbose = TRUE)
manitowoc_weather_2019 <- lcd(station = "72645594897", year = 2019, verbose = TRUE)
manitowoc_weather_2020 <- lcd(station = "72645594897", year = 2020, verbose = TRUE)
manitowoc_weather_2021 <- lcd(station = "72645594897", year = 2021, verbose = TRUE)
```

Data Processing

`rbind` allows us to bind rows together, one on top of the other. Data frames must have the same number of columns with the same names.

The `dplyr` package within `tidyverse` has a term called a pipe, designated as %>%. In coding language, the pipe means "then," allowing you to sequentially run functions on a given data frame.

Other wrangling functions we will use (feel free to add notes here or comment in the code):

- `select`
- `drop_na`
- `separate`
- `mutate`
- `filter`
- `group_by`
- `summarise`

```
# combine all years
manitowoc_weather <- rbind(manitowoc_weather_2012, manitowoc_weather_2013,
                           manitowoc_weather_2014, manitowoc_weather_2015,
                           manitowoc_weather_2016, manitowoc_weather_2017,
                           manitowoc_weather_2018, manitowoc_weather_2019,
                           manitowoc_weather_2020, manitowoc_weather_2021)
# examine the dataset
colnames(manitowoc_weather)
```

```
##   [1] "station"
##   [2] "date"
##   [3] "latitude"
##   [4] "longitude"
##   [5] "elevation"
##   [6] "name"
##   [7] "report_type"
##   [8] "source"
##   [9] "hourlyaltimetersetting"
##  [10] "hourlydewpointtemperature"
##  [11] "hourlydrybulbtemperature"
##  [12] "hourlyprecipitation"
##  [13] "hourlypresentweathertype"
##  [14] "hourlypressurechange"
##  [15] "hourlypressuretendency"
##  [16] "hourlyrelativehumidity"
##  [17] "hourlyskyconditions"
##  [18] "hourlysealevelpressure"
##  [19] "hourlystationpressure"
##  [20] "hourlyvisibility"
##  [21] "hourlywetbulbtemperature"
##  [22] "hourlywinddirection"
##  [23] "hourlywindgustspeed"
```

```
##  [24] "hourlywindspeed"
##  [25] "sunrise"
##  [26] "sunset"
##  [27] "dailyaveragedewpointtemperature"
##  [28] "dailyaveragedrybulbtemperature"
##  [29] "dailyaveragerelativehumidity"
##  [30] "dailyaveragesealevelpressure"
##  [31] "dailyaveragestationpressure"
##  [32] "dailyaveragewetbulbtemperature"
##  [33] "dailyaveragewindspeed"
##  [34] "dailycoolingdegreedays"
##  [35] "dailydeparturefromnormalaveragetemperature"
##  [36] "dailyheatingdegreedays"
##  [37] "dailymaximumdrybulbtemperature"
##  [38] "dailyminimumdrybulbtemperature"
##  [39] "dailypeakwinddirection"
##  [40] "dailypeakwindspeed"
##  [41] "dailyprecipitation"
##  [42] "dailysnowdepth"
##  [43] "dailysnowfall"
##  [44] "dailysustainedwinddirection"
##  [45] "dailysustainedwindspeed"
##  [46] "dailyweather"
##  [47] "monthlyaveragerh"
##  [48] "monthlydayswithgt001precip"
##  [49] "monthlydayswithgt010precip"
##  [50] "monthlydayswithgt32temp"
##  [51] "monthlydayswithgt90temp"
##  [52] "monthlydayswithlt0temp"
##  [53] "monthlydayswithlt32temp"
##  [54] "monthlydeparturefromnormalaveragetemperature"
##  [55] "monthlydeparturefromnormalcoolingdegreedays"
##  [56] "monthlydeparturefromnormalheatingdegreedays"
##  [57] "monthlydeparturefromnormalmaximumtemperature"
##  [58] "monthlydeparturefromnormalminimumtemperature"
##  [59] "monthlydeparturefromnormalprecipitation"
##  [60] "monthlydewpointtemperature"
##  [61] "monthlygreatestprecip"
##  [62] "monthlygreatestprecipdate"
##  [63] "monthlygreatestsnowdepth"
##  [64] "monthlygreatestsnowdepthdate"
##  [65] "monthlygreatestsnowfall"
##  [66] "monthlygreatestsnowfalldate"
##  [67] "monthlymaxsealevelpressurevalue"
##  [68] "monthlymaxsealevelpressurevaluedate"
##  [69] "monthlymaxsealevelpressurevaluetime"
##  [70] "monthlymaximumtemperature"
##  [71] "monthlymeantemperature"
##  [72] "monthlyminsealevelpressurevalue"
##  [73] "monthlyminsealevelpressurevaluedate"
##  [74] "monthlyminsealevelpressurevaluetime"
##  [75] "monthlyminimumtemperature"
##  [76] "monthlysealevelpressure"
##  [77] "monthlystationpressure"
```

```
##  [78] "monthlytotalliquidprecipitation"
##  [79] "monthlytotalsnowfall"
##  [80] "monthlywetbulb"
##  [81] "awnd"
##  [82] "cdsd"
##  [83] "cldd"
##  [84] "dsnw"
##  [85] "hdsd"
##  [86] "htdd"
##  [87] "normalscoolingdegreeday"
##  [88] "normalsheatingdegreeday"
##  [89] "shortdurationenddate005"
##  [90] "shortdurationenddate010"
##  [91] "shortdurationenddate015"
##  [92] "shortdurationenddate020"
##  [93] "shortdurationenddate030"
##  [94] "shortdurationenddate045"
##  [95] "shortdurationenddate060"
##  [96] "shortdurationenddate080"
##  [97] "shortdurationenddate100"
##  [98] "shortdurationenddate120"
##  [99] "shortdurationenddate150"
## [100] "shortdurationenddate180"
## [101] "shortdurationprecipitationvalue005"
## [102] "shortdurationprecipitationvalue010"
## [103] "shortdurationprecipitationvalue015"
## [104] "shortdurationprecipitationvalue020"
## [105] "shortdurationprecipitationvalue030"
## [106] "shortdurationprecipitationvalue045"
## [107] "shortdurationprecipitationvalue060"
## [108] "shortdurationprecipitationvalue080"
## [109] "shortdurationprecipitationvalue100"
## [110] "shortdurationprecipitationvalue120"
## [111] "shortdurationprecipitationvalue150"
## [112] "shortdurationprecipitationvalue180"
## [113] "rem"
## [114] "backupdirection"
## [115] "backupdistance"
## [116] "backupdistanceunit"
## [117] "backupelements"
## [118] "backupelevation"
## [119] "backupequipment"
## [120] "backuplatitude"
## [121] "backuplongitude"
## [122] "backupname"
## [123] "windequipmentchangedate"
```

```
View(manitowoc_weather)

# select columns for general info and daily conditions
manitowoc_weather_daily <- manitowoc_weather %>%
  #
  select(station:name, dailyaveragestationpressure, dailyaveragewindspeed,
         dailymaximumdrybulbtemperature, dailyminimumdrybulbtemperature,
```

```
        dailyprecipitation, dailyweather)
View(manitowoc_weather_daily)

# Precipitation: T indicates trace. Set to zero.
# Precipitation: Blank represents times that were not sampled
# Precipitation: s indicates suspect. Set to NA.
manitowoc_weather_daily$dailyprecipitation[manitowoc_weather_daily$dailyprecipitation == "T"] <- 0 #
manitowoc_weather_daily$dailyprecipitation <- as.numeric(manitowoc_weather_daily$dailyprecipitation) #
summary(manitowoc_weather_daily$dailyprecipitation)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00    0.07    0.01    3.03  130223
```

```
# Temperature: air temperature = dry bulb temperature.
# Temperature: s indicates suspect. Set to NA.
manitowoc_weather_daily$dailymaximumdrybulbtemperature <-
  as.numeric(manitowoc_weather_daily$dailymaximumdrybulbtemperature)
manitowoc_weather_daily$dailyminimumdrybulbtemperature <-
  as.numeric(manitowoc_weather_daily$dailyminimumdrybulbtemperature)

# remove rows that do not contain precipitation data
# access date information
manitowoc_weather_daily <- manitowoc_weather_daily %>%
  #
  drop_na(dailyprecipitation) %>%
  #
  separate(col = date, into = c("Date", "Time"), sep = "T")

summary(manitowoc_weather_daily)
```

```
##     station              Date               Time              latitude
##  Min.   :7.265e+10   Length:2341        Length:2341        Min.   :44.13
##  1st Qu.:7.265e+10   Class :character   Class :character   1st Qu.:44.13
##  Median :7.265e+10   Mode  :character   Mode  :character   Median :44.13
##  Mean   :7.265e+10                                         Mean   :44.13
##  3rd Qu.:7.265e+10                                         3rd Qu.:44.13
##  Max.   :7.265e+10                                         Max.   :44.13
##
##    longitude        elevation         name
##  Min.   :-87.67   Min.   :198.4   Length:2341
##  1st Qu.:-87.67   1st Qu.:198.4   Class :character
##  Median :-87.67   Median :198.4   Mode  :character
##  Mean   :-87.67   Mean   :198.4
##  3rd Qu.:-87.67   3rd Qu.:198.4
##  Max.   :-87.67   Max.   :198.4
##
##  dailyaveragestationpressure dailyaveragewindspeed
##  Min.   :28.53               Min.   : 0.000
##  1st Qu.:29.18               1st Qu.: 6.000
##  Median :29.31               Median : 8.200
##  Mean   :29.31               Mean   : 8.608
##  3rd Qu.:29.45               3rd Qu.:10.800
##  Max.   :30.08               Max.   :24.300
##  NA's   :112                 NA's   :87
```

```
##  dailymaximumdrybulbtemperature dailyminimumdrybulbtemperature
##  Min.   :-12.00                 Min.   :-23.00
##  1st Qu.: 36.00                 1st Qu.: 23.00
##  Median : 50.00                 Median : 34.00
##  Mean   : 51.59                 Mean   : 34.92
##  3rd Qu.: 70.00                 3rd Qu.: 51.00
##  Max.   : 92.00                 Max.   : 76.00
##  NA's   :398                    NA's   :510
##  dailyprecipitation dailyweather
##  Min.   :0.00000    Length:2341
##  1st Qu.:0.00000    Class :character
##  Median :0.00000    Mode  :character
##  Mean   :0.07392
##  3rd Qu.:0.01000
##  Max.   :3.03000
##
```

```
str(manitowoc_weather_daily)
```

```
## tibble [2,341 x 13] (S3: tbl_df/tbl/data.frame)
##  $ station                       : num [1:2341] 7.26e+10 7.26e+10 7.26e+10 7.26e+10 7.26e+10 ...
##  $ Date                          : chr [1:2341] "2012-01-02" "2012-01-03" "2012-01-04" "2012-01-05"
##  $ Time                          : chr [1:2341] "23:59:00" "23:59:00" "23:59:00" "23:59:00" ...
##  $ latitude                      : num [1:2341] 44.1 44.1 44.1 44.1 44.1 ...
##  $ longitude                     : num [1:2341] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ elevation                     : num [1:2341] 198 198 198 198 198 ...
##  $ name                          : chr [1:2341] "MANITOWOC, WI US" "MANITOWOC, WI US" "MANITOWOC, WI
##  $ dailyaveragestationpressure   : num [1:2341] 29.4 29.5 29.2 29.1 28.9 ...
##  $ dailyaveragewindspeed         : num [1:2341] 17.7 9.4 9.9 7 10.9 8.7 6.5 13.6 6.4 3.4 ...
##  $ dailymaximumdrybulbtemperature: num [1:2341] NA NA NA NA NA NA NA NA NA NA ...
##  $ dailyminimumdrybulbtemperature: num [1:2341] NA NA NA NA NA NA NA NA NA NA ...
##  $ dailyprecipitation            : num [1:2341] 0 0 0 0 0 0 0 0 0 0 ...
##  $ dailyweather                  : chr [1:2341] "" "" "" "" ...
```

```
# change date to date
manitowoc_weather_daily$Date <- as.Date(manitowoc_weather_daily$Date, format = "%Y-%m-%d")

# add month and year columns
# may want to use water year, not calendar year
# notice data were not supplied prior to October 2013. Let's remove dates prior to 2014.
manitowoc_weather_daily <- manitowoc_weather_daily %>%
  #
  mutate(Year = year(Date),
         WaterYear = water_year(Date, origin = "usgs"),
         Month = month(Date)) %>%
  #
  filter(Year > 2013)
```

Interpolate missing values

We restricted our daily data to dates that have precipitation present. However, there are NA values in our temperature dataset. There are a few ways to interpolate missing data, *if deemed appropriate for your analysis needs.*

A few common interpolation methods for temporal data:

- **Piecewise constant, aka nearest neighbor**: assign same value as nearest data point
- **Linear**: "connect the dots"
- **Spline**: polynomial functions, results in curved interpolation line

When working with spatial data, additional interpolation methods are commonly used, including **kriging**, **bilinear**, and **cubic convolutions**.

If we want to fill in missing temperature data, it is a reasonable assumption that a missing daily temperature could be well-approximated by the average of the temperature on the day before and the day after. So, a linear interpolation would make sense here.

Note: it may be beneficial to designate which values in your dataset were measured vs. interpolated. A few options to do this are (a) create a new column in your dataset for your variable that includes both measured and interpolated values (leaving the original column as-is), or (b) create a new column in your dataset designating whether the value in the variable's column is interpolated or not (0/1 or yes/no).

```
manitowoc_weather_daily$dailymaximumdrybulbtemperature <- na.approx(manitowoc_weather_daily$dailymaximum
manitowoc_weather_daily$dailyminimumdrybulbtemperature <- na.approx(manitowoc_weather_daily$dailyminimum
```

Data Aggregation

From our daily data, let's aggregate those data up to monthly and annual conditions. Precipitation will be summed to generate monthly and yearly totals, and temperatures will be averaged. The mean function automatically generates a blank if an NA is supplied as part of the calculation, so

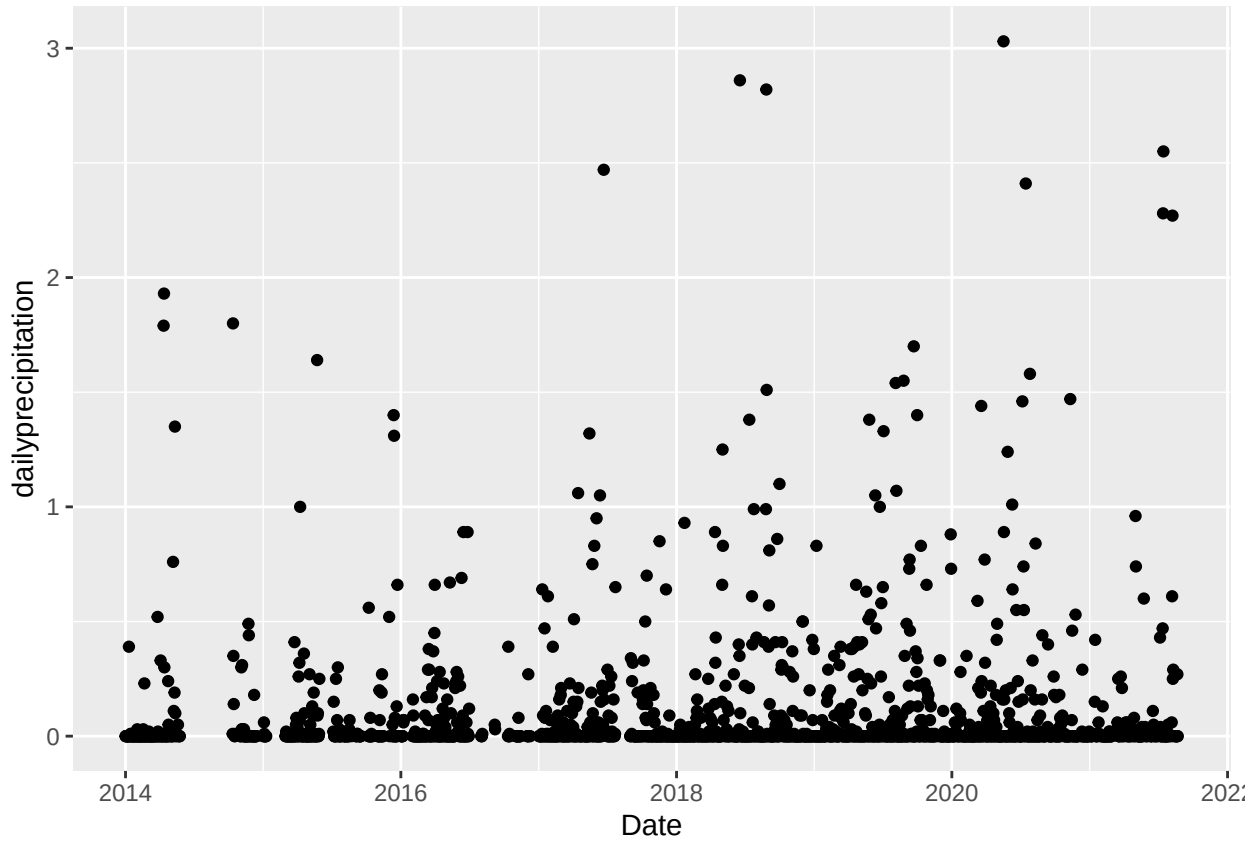
```
manitowoc_preciptemp_monthly <- manitowoc_weather_daily %>%
  #
  group_by(WaterYear, Month) %>%
  #
  summarise(monthlyprecipitation = sum(dailyprecipitation),
            avgmonthlymaxtemp = mean(dailymaximumdrybulbtemperature, na.rm = TRUE),
            avgmonthlymintemp = mean(dailyminimumdrybulbtemperature, na.rm = TRUE))

# Your turn. Calculate annual precipitation by water year, using a pipe
manitowoc_preciptemp_annual <- manitowoc_weather_daily
```

Exploratory visualization

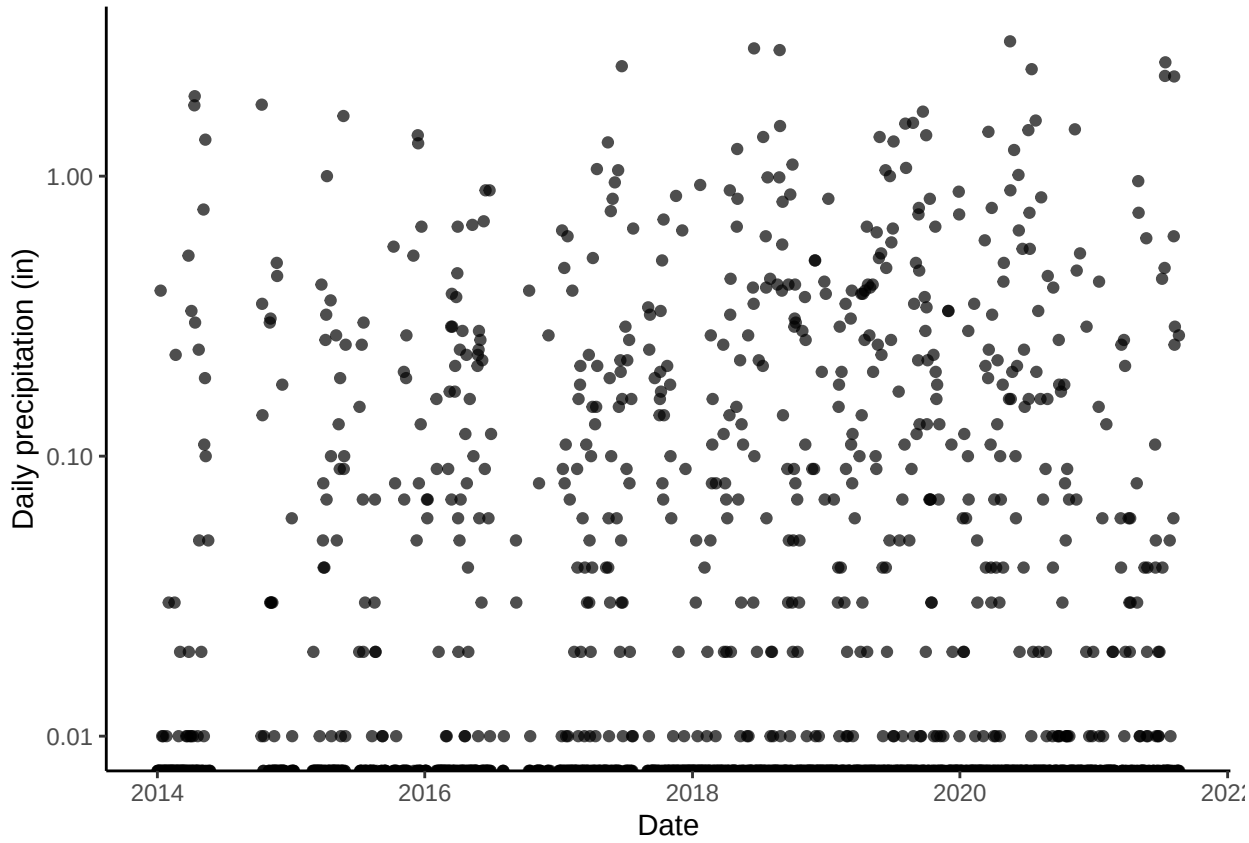
Time series

```
# basic time series plot
ggplot(manitowoc_weather_daily, aes(x = Date, y = dailyprecipitation)) +
  geom_point()
```




```
# improve the visualization
ggplot(manitowoc_weather_daily, aes(x = Date, y = dailyprecipitation)) +
  geom_point(alpha = 0.7) +
  scale_y_log10() +
  labs(y = "Daily precipitation (in)") +
  theme_classic()
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```



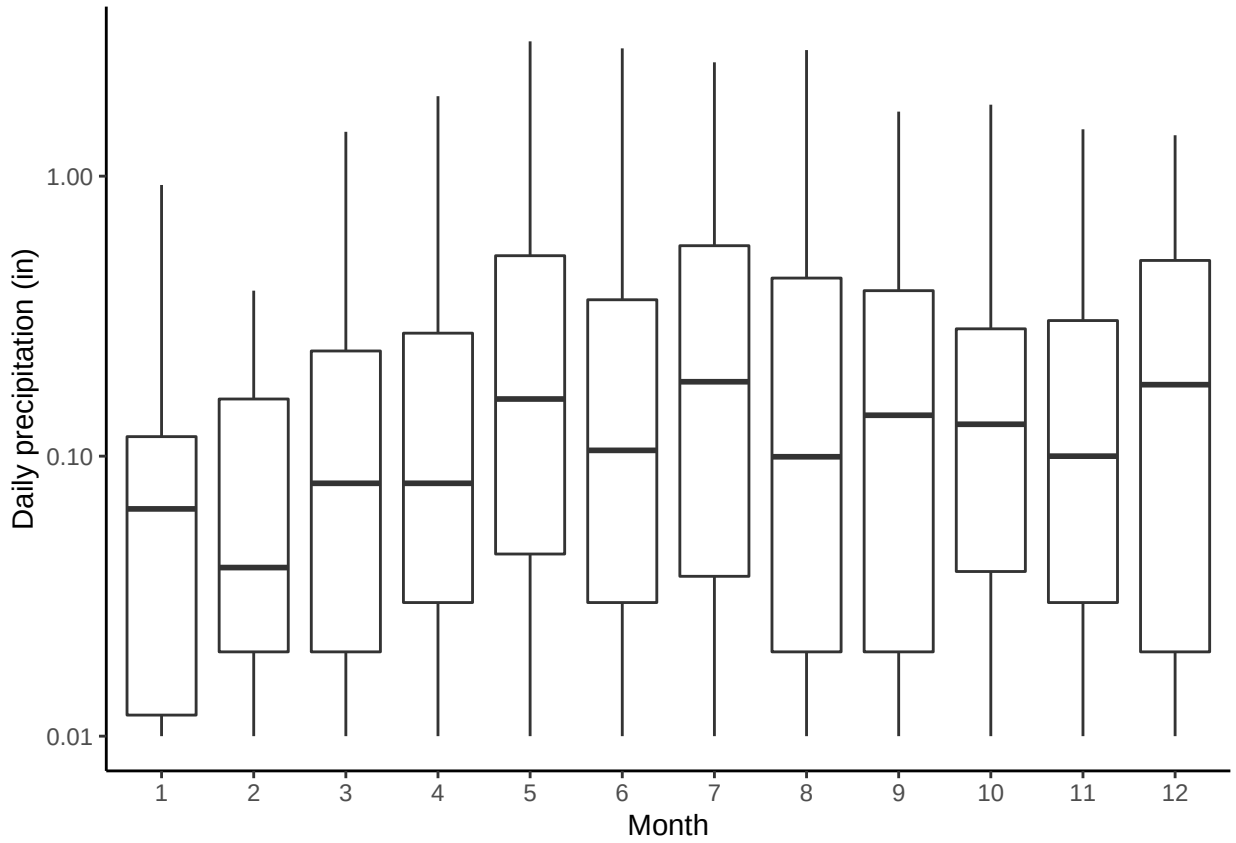

```
# your turn. create a time series of daily maximum temperatures.
# does it make sense to log the y axis for temperature?
```

Monthly conditions

```
ggplot(manitowoc_weather_daily, aes(x = as.factor(Month), y = dailyprecipitation)) +
  geom_boxplot() +
  scale_y_log10() +
  #facet_wrap(vars(WaterYear), ncol = 1) +
  labs(x = "Month", y = "Daily precipitation (in)") +
  theme_classic()
```

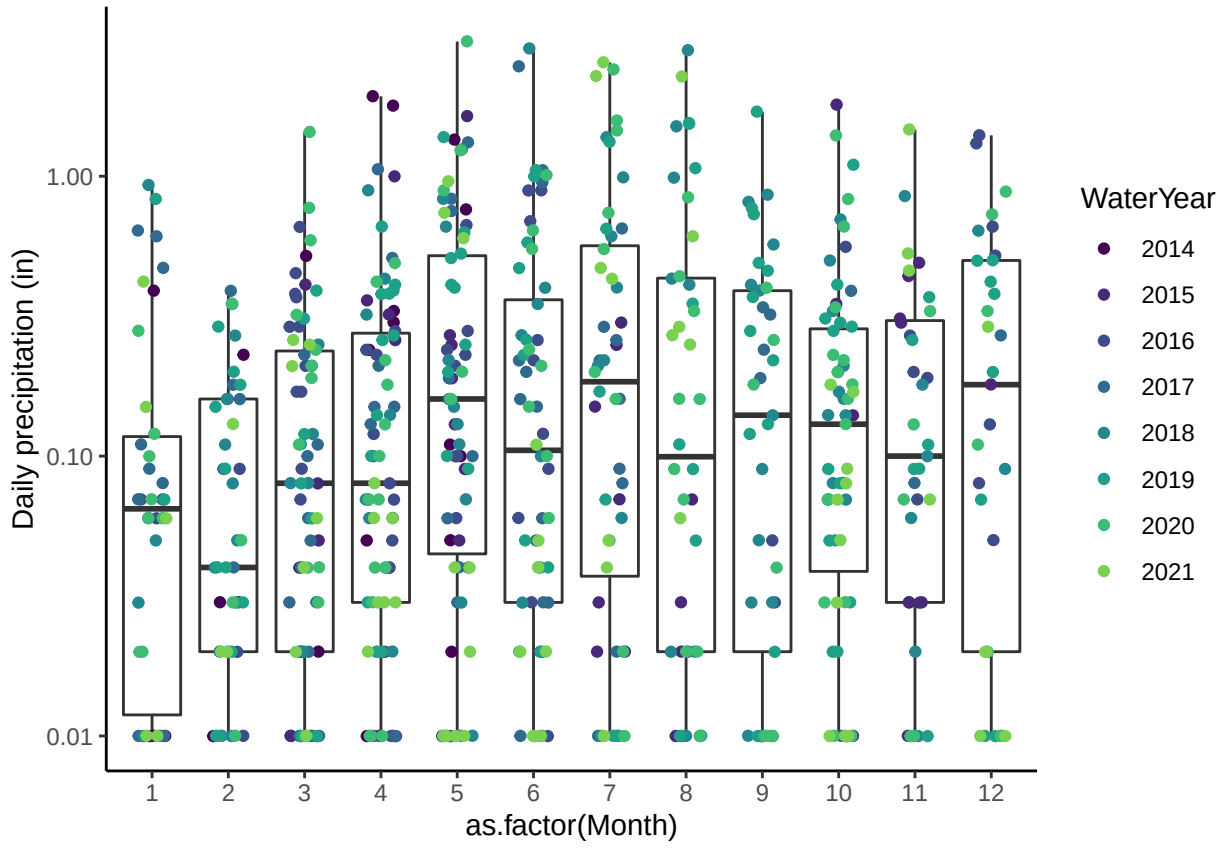
```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 1398 rows containing non-finite values (stat_boxplot).
```




```
ggplot(manitowoc_weather_daily, aes(x = as.factor(Month), y = dailyprecipitation)) +
  geom_boxplot() +
  geom_jitter(aes(color = WaterYear), width = 0.2) +
  scale_y_log10() +
  labs(y = "Daily precipitation (in)") +
  scale_color_viridis_d(end = 0.8) +
  theme_classic()
```

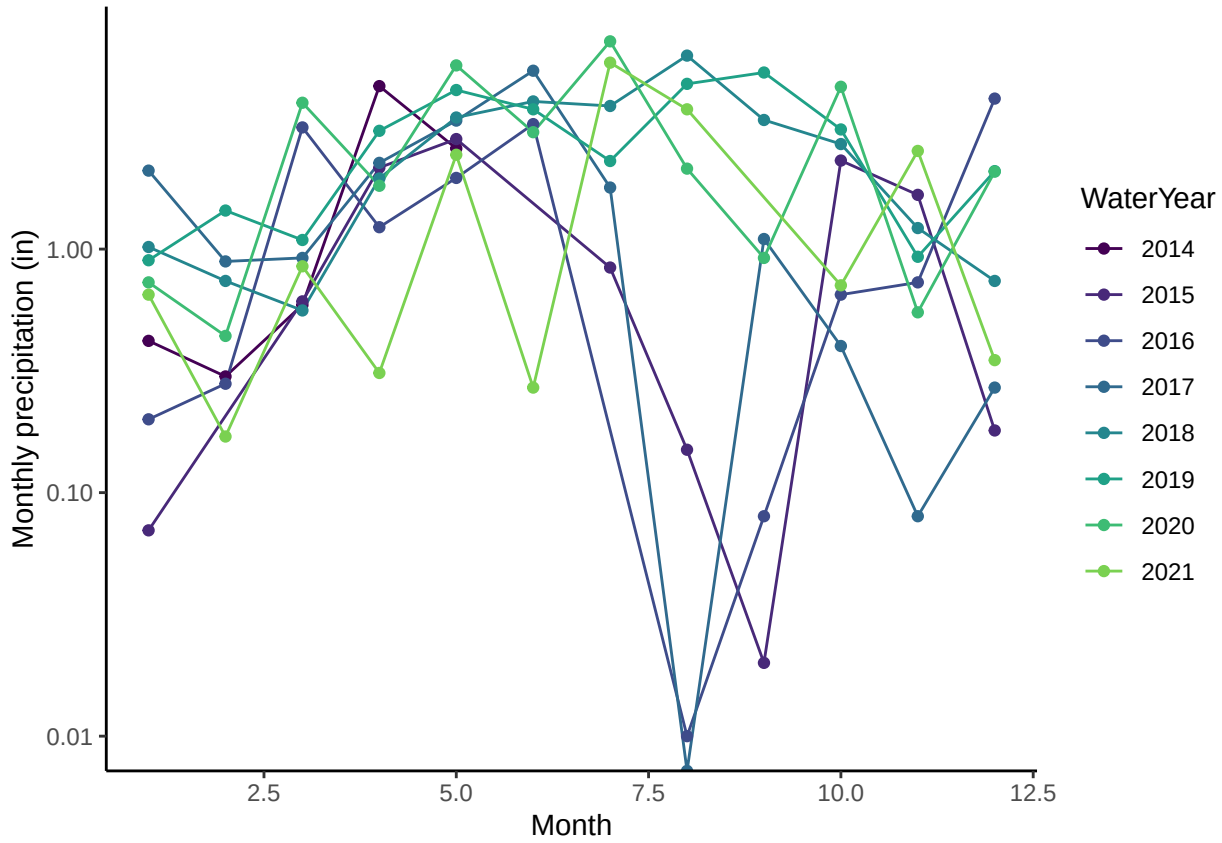
```
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Removed 1398 rows containing non-finite values (stat_boxplot).
## Warning: Removed 1398 rows containing missing values (geom_point).
```




```
ggplot(manitowoc_preciptemp_monthly, aes(x = Month, y = monthlyprecipitation, color = WaterYear)) +
  geom_line() +
  geom_point() +
  scale_y_log10() +
  labs(y = "Monthly precipitation (in)") +
  scale_color_viridis_d(end = 0.8) +
  theme_classic()
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```




```
# your turn. plot annual precipitation by water year.
# what style of graph makes sense?
```

Data Management

Saving datasets

General rules:

1. Save processed datasets separately from raw data
2. Use informative file names that are easily sorted and organized in a folder
3. No spaces in file names

```
write.csv(manitowoc_weather_daily, file = "./Data/Processed/Manitowoc_weather_daily_2012-2021.csv",
          row.names = FALSE)

write.csv(manitowoc_preciptemp_monthly, file = "./Data/Processed/Manitowoc_weather_monthly_2014-2021.csv
          row.names = FALSE)
```

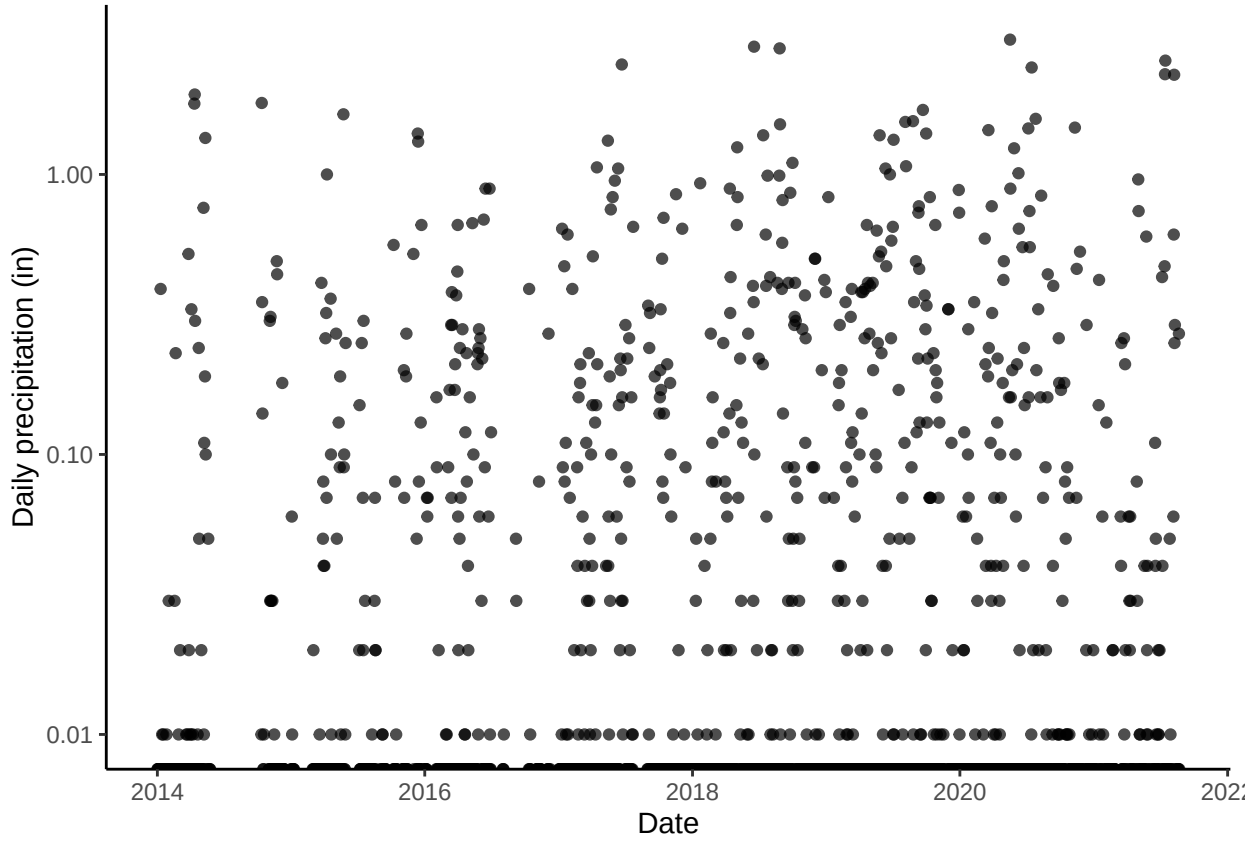
Saving graphs

In RStudio, you can manually export graphs by saving as image, PDF, or copy-paste. However, the image resolution is poor by this method, and the method itself is not reproducible. Saving your graphs using code is preferable.

`ggsave` automatically saves the current plot if you don't specify which plot. Alternatively, you can name a plot and save that named plot at any point (does not need to be plotted at the time of saving). Height and width are automatically in inches, and the text size scales accordingly (smaller dimensions = larger text).

```
ggplot(manitowoc_weather_daily, aes(x = Date, y = dailyprecipitation)) +
  geom_point(alpha = 0.7) +
  scale_y_log10() +
  labs(y = "Daily precipitation (in)") +
  theme_classic()
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```




```
# ggsave("./Output/manitowoc_precipitation_timeseries.jpg",
#        height = 4, width = 5)

precip_monthly_boxplot <-
ggplot(manitowoc_weather_daily, aes(x = as.factor(Month), y = dailyprecipitation)) +
  geom_boxplot() +
  scale_y_log10() +
  labs(x = "Month", y = "Daily precipitation (in)") +
  theme_classic()
# ggsave(precip_monthly_boxplot, "./Output/manitowoc_precipitation_monthlyboxplot.jpg",
#        height = 4, width = 5)
```