Sagehen Groundwater Data Processing

Jennifer Natali, with help from chatgpt for learning R, dplyr, stats

2024 October 24

Load and Summarize Response Data

1. Upload data (.csv file)

- groundwater level
- TODO: plant greenness from phenopix
- TODO: discharge?

2. Examine the following properties:

- length and frequency of the time series
- completeness of each time series
- descriptive statistics
  - basics: mean, CV, ACF for each variable
  - normality: histogram, qqplot, skewness, kurtosis

```
# Load libraries
library(dplyr)
library(tidyr)
library(astsa)
library(lubridate)
library(moments) # for skewness and kurtosis testing

# Setup directories and filepaths
home_dir='/Volumes/SANDISK_SSD_G40/GoogleDrive/GitHub/'
repository_dir = paste(home_dir,'sagehen_meadows/', sep='')
groundwater_data_dir = 'data/field_observations/groundwater/biweekly_manual/'
groundwater_filepath = paste(repository_dir, groundwater_data_dir,
                             'groundwater_biweekly_FULL.csv', sep='')
observation_filepath = paste(repository_dir, groundwater_data_dir,
                             'groundwater_biweekly_observation_spacing.csv',
                             sep='')

# Load groundwater data
groundwater <- read.csv(groundwater_filepath)

# ---TODO: Add "greater_than" data to increase completeness (for now)
# ---TODO: Consider adding data from Kirchner 2006-2008 B+D xect
# ---TODO: Get data from other (not my) transducers for 2018-2024?

# Manage dates and times
groundwater$timestamp <- ymd_hms(groundwater$timestamp)

# Check timestamp formatting
str(groundwater$timestamp)
```

```
##  POSIXct[1:1359], format: "2018-06-01 07:45:00" "2018-06-18 08:32:00" "2018-06-30 08:55:00" ...
```

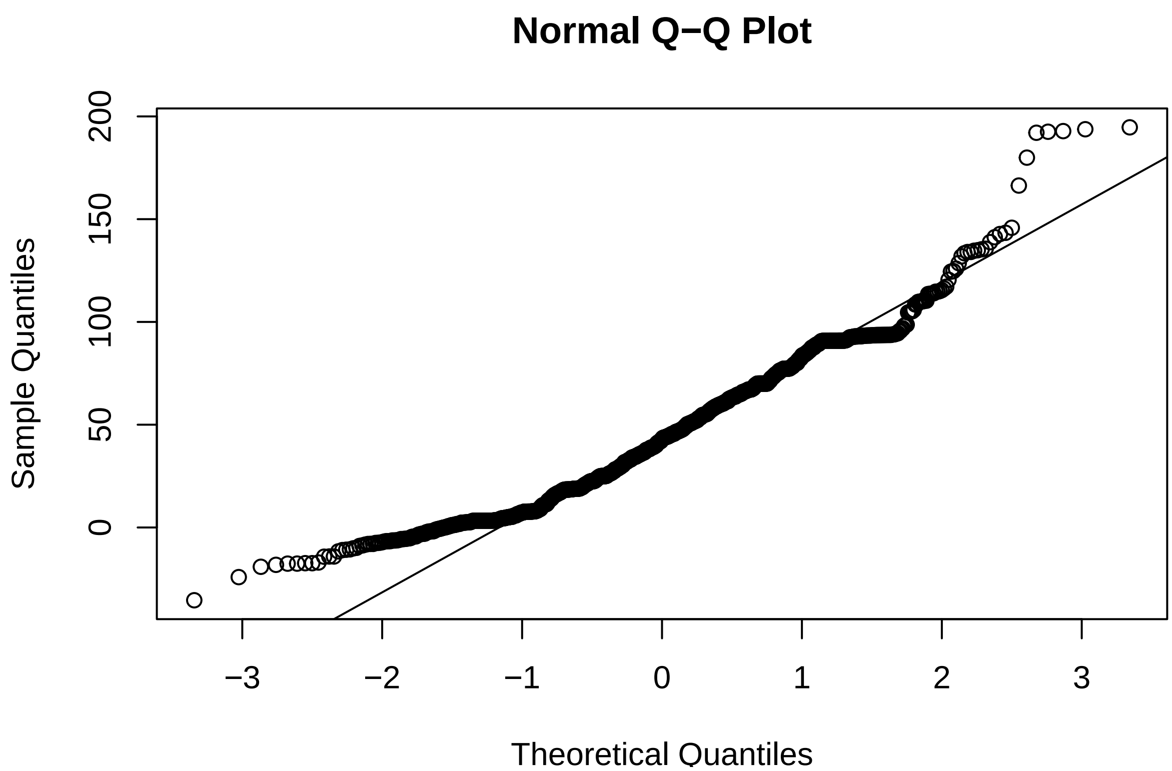
```r
# Create columns for date and isoweek (starts on Monday)
groundwater <- groundwater %>% mutate(
  date = as.Date(timestamp),
  year = year(timestamp),
  isoweek = isoweek(date),
  day_of_year = yday(date))

# summarize the full times series
summary(groundwater)
```

```
##    well_id            timestamp                      ground_to_water_cm
##  Length:1359        Min.   :2018-05-31 08:30:00.0   Min.   :-35.41
##  Class :character   1st Qu.:2018-10-01 07:53:00.0   1st Qu.: 18.47
##  Mode  :character   Median :2019-08-19 08:00:00.0   Median : 42.92
##                     Mean   :2019-12-31 09:46:14.7   Mean   : 45.14
##                     3rd Qu.:2021-06-26 16:30:00.0   3rd Qu.: 69.38
##                     Max.   :2021-11-14 10:14:00.0   Max.   :194.67
##                                                     NA's   :151
##       date                 year         isoweek      day_of_year
##  Min.   :2018-05-31   Min.   :2018   Min.   :20.0   Min.   :140.0
##  1st Qu.:2018-10-01   1st Qu.:2018   1st Qu.:27.0   1st Qu.:185.5
##  Median :2019-08-19   Median :2019   Median :31.0   Median :217.0
##  Mean   :2019-12-31   Mean   :2019   Mean   :31.9   Mean   :221.5
##  3rd Qu.:2021-06-26   3rd Qu.:2021   3rd Qu.:37.0   3rd Qu.:259.0
##  Max.   :2021-11-14   Max.   :2021   Max.   :46.0   Max.   :322.0
##
```

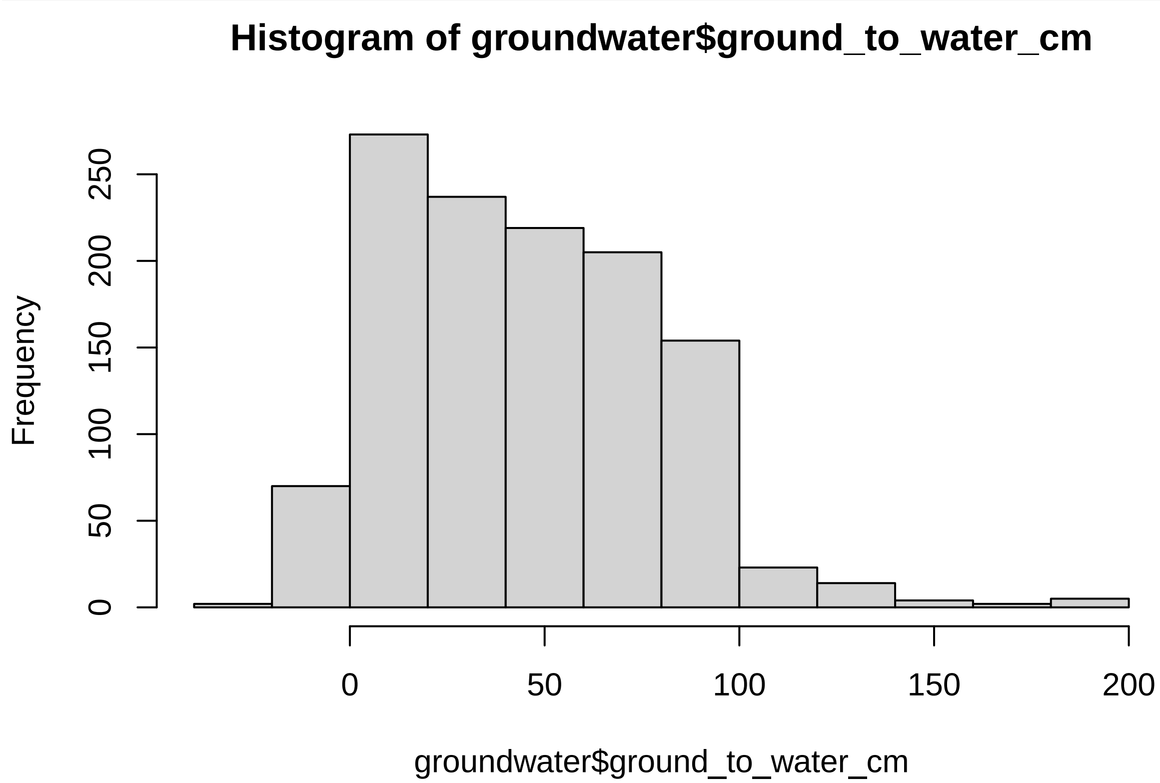
```r
nrow_groundwater_orig <- nrow(groundwater)

# explore the distribution of the data
# NOTE: does NOT need to be normally distributed for MAR/MARSS models

# shapiro-wilk test; data is likely non-normal if p-value < 0.05
shapiro.test(groundwater$ground_to_water_cm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  groundwater$ground_to_water_cm
## W = 0.96332, p-value < 2.2e-16
```

```r
# results: p << 0.05, data is non-normal

# Q-Q plots; data is normal if falls on a straight line
qqnorm(groundwater$ground_to_water_cm)
qqline(groundwater$ground_to_water_cm)
```




```r
# results: mostly normal but some outliers

# Histogram; check for bell-shaped curve
hist(groundwater$ground_to_water_cm)
```



```r
# Skewness; test if near 0 (symmetric), >0 (positive skew), <0 (neg skew)
skewness(groundwater$ground_to_water_cm, na.rm = TRUE)
```

```
## [1] 0.6387792
```

```r
# result: 0.83; positive skewed, >0.5 so moderately skewed

# Kurtosis; test for heavy tails
# (if ~3 normal, if >3 heavy tails + sharp peak, if <3 light tails, flat peak)
kurtosis(groundwater$ground_to_water_cm, na.rm = TRUE)
```

```
## [1] 3.655301
```

```r
# result 4.5; heavy tail and sharp peak

# summarize the span of full time series by date
groundwater %>% summarize(
  start_date = min(date, na.rm=TRUE),
  stop_date = max(date, na.rm=TRUE),
  timespan = difftime(stop_date, start_date, units="days"),
  unique_dates_count = n_distinct(date)
)
```

```
##   start_date  stop_date  timespan unique_dates_count
## 1 2018-05-31 2021-11-14 1263 days                191
```

```r
# basic descriptive statistics across all groundwater readings: mean, CV, ACF
groundwater %>%
  summarize(
    mean_value = mean(ground_to_water_cm, na.rm = TRUE),
    sd_value = sd(ground_to_water_cm, na.rm = TRUE),
    var_value = var(ground_to_water_cm, na.rm = TRUE),
    cv_value = 100 * sd_value / mean_value
  )
```

```
##   mean_value sd_value var_value cv_value
## 1   45.14073    34.79  1210.344 77.07009
```

```r
# basic descriptive statistics by groupings: mean, CV, ACF for each variable
groundwater_summary_by_date <- groundwater %>%
  group_by(date) %>%
  summarise(
    mean_value = mean(ground_to_water_cm, na.rm = TRUE),
    sd_value = sd(ground_to_water_cm, na.rm = TRUE),
    var_value = var(ground_to_water_cm, na.rm = TRUE)
  )

groundwater_summary_by_well <- groundwater %>%
  group_by(well_id) %>%
  summarise(
    mean_value = mean(ground_to_water_cm, na.rm = TRUE),
    sd_value = sd(ground_to_water_cm, na.rm = TRUE),
    var_value = var(ground_to_water_cm, na.rm = TRUE)
  )
groundwater_summary_by_well
```

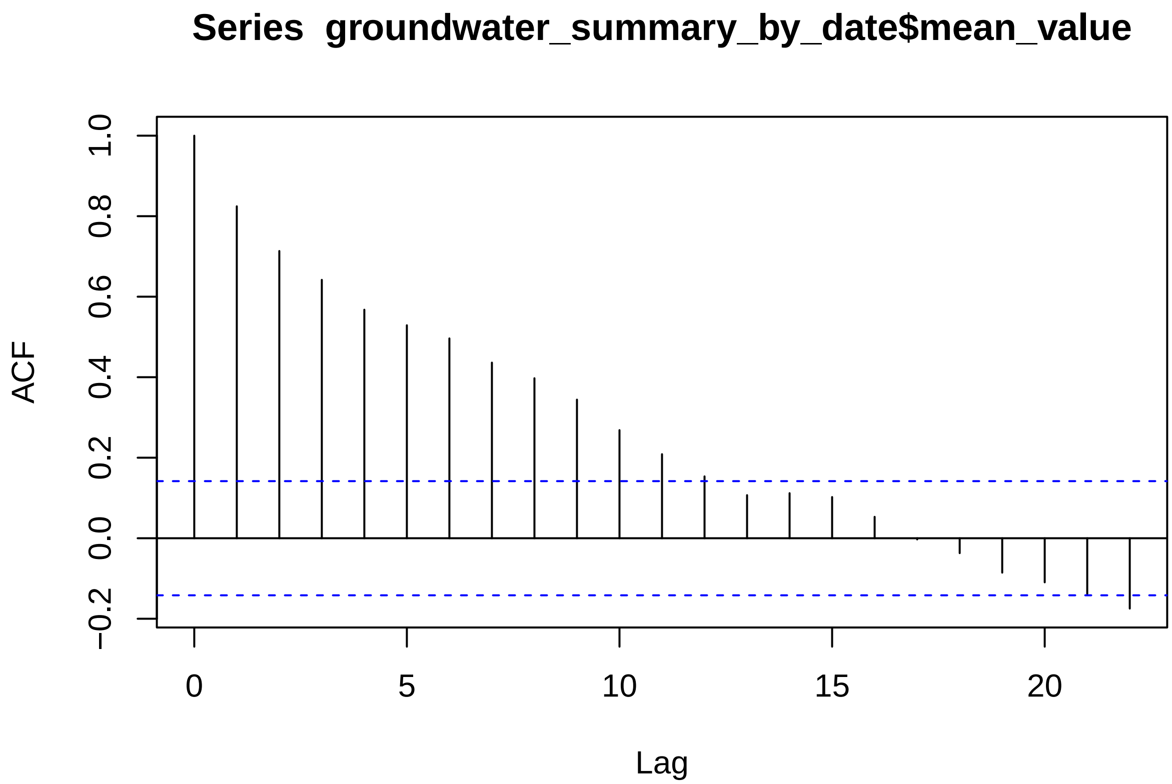
```
## # A tibble: 54 x 4
```

```
##    well_id  mean_value sd_value var_value
##    <chr>         <dbl>    <dbl>     <dbl>
##  1 EEF-1          14.3    22.9      523.
##  2 EER-1          34.1    10.8      117.
##  3 EET-1          59.9    32.8     1076.
##  4 EET-2         101.     41.2     1699.
##  5 EET-XB4S       31.6    20.0      400.
##  6 EFF-XA1N       40.1    11.3      127.
##  7 EFF-XA2N       27.0    11.6      134.
##  8 EFF-XB7S       30.5    36.8     1357.
##  9 EFR-XB1S       45.2     4.96      24.6
## 10 EFR-XB2N       28.7    11.3      127.
## # i 44 more rows
```

```r
groundwater_summary_by_well_year <- groundwater %>%
  mutate(year = year(timestamp)) %>%
  group_by(well_id, year) %>%
  summarise(
    mean_value = mean(ground_to_water_cm, na.rm = TRUE),
    sd_value = sd(ground_to_water_cm, na.rm = TRUE),
    var_value = var(ground_to_water_cm, na.rm = TRUE),
    .groups = "keep"
  )
groundwater_summary_by_well_year
```

```
## # A tibble: 141 x 5
## # Groups:   well_id, year [141]
##    well_id  year mean_value sd_value var_value
##    <chr>   <dbl>      <dbl>    <dbl>     <dbl>
##  1 EEF-1    2018       7.20     4.51      20.3
##  2 EEF-1    2019       7.34    12.7      160.
##  3 EEF-1    2021      61.0     29.8      888.
##  4 EER-1    2018      34.5      8.50      72.3
##  5 EER-1    2019      27.6     12.4      155.
##  6 EER-1    2021      39.1      8.76      76.7
##  7 EET-1    2018      65.4     31.7     1007.
##  8 EET-1    2019      57.0     41.7     1736.
##  9 EET-1    2021      56.4     31.1      968.
## 10 EET-2    2019     107.      20.4      418.
## # i 131 more rows
```

```r
# Trying to get acf, but not sure if this summarized data means anything
acf(groundwater_summary_by_date$mean_value)
```




```r
## ORGANIZE BY AN EVEN TIMESTEP
# summarize the weekly data
groundwater_weekly_summary <- groundwater %>%
  group_by(isoweek) %>%
  summarize(
      n_week = n()   # Number of entries in each week
  )
groundwater_weekly_summary
```

```
## # A tibble: 25 x 2
##    isoweek n_week
##      <dbl>  <int>
##  1      20     27
##  2      22     84
##  3      23     24
##  4      24     58
##  5      25     67
##  6      26     72
##  7      27     85
##  8      28     10
##  9      29     97
## 10      30    105
## # i 15 more rows
```

```r
# summarize spacing of the full time series
unique_observations <- groundwater %>%
  select(-well_id, -ground_to_water_cm, -timestamp) %>%
  distinct(date, .keep_all=TRUE) %>%
  group_by(year) %>%
  arrange(date) %>%
```

```r
  mutate(day_diff = as.numeric(difftime(lead(date), date, units="days")))

unique_observations %>% summarize(
  max_days = max(day_diff, na.rm=TRUE),
  mean_days = mean(day_diff, na.rm=TRUE),
)
```

```
## # A tibble: 3 x 3
##    year max_days mean_days
##   <dbl>    <dbl>     <dbl>
## 1  2018       35      3.23
## 2  2019       16      1.16
## 3  2021       39     13.7
```

```r
# evaluate how many observation dates are "close" & if they're in the same week
close_threshold = 3 # consider observations "close" if <3 days apart
same_week_count <- unique_observations %>%
  mutate(next_isoweek = lead(isoweek)) %>%
  filter(day_diff < close_threshold) %>%
  summarise(
    number_close_days = n(),
    number_same_week = sum(isoweek == next_isoweek)
  ) %>%
  mutate(
    percent_close = number_same_week / number_close_days * 100
  )
same_week_count
```

```
## # A tibble: 3 x 4
##    year number_close_days number_same_week percent_close
##   <dbl>             <int>            <int>         <dbl>
## 1  2018                46               40          87.0
## 2  2019               120              102          85
## 3  2021                 2                1          50
```

```r
# filter measurements, only before AM threshold
am_time_limit <- 10
groundwater_filter_by_time <- groundwater %>%
  filter(hour(timestamp) > am_time_limit)

# number of observations removed by time limit
nrow(groundwater_filter_by_time)
```

```
## [1] 141
```

```r
groundwater <- groundwater %>%
  filter(hour(timestamp) <= am_time_limit)

# # number of observations that'll be lost from
# # filtering for duplicate entries (same well, same week)
# groundwater_filter_duplicates <- groundwater %>%
#   group_by(well_id, year, isoweek) %>%
#   filter(n() > 1) %>%
#   ungroup()
# nrow(groundwater_filter_duplicates)
#
```

```r
# # remove duplicate entries (same well, same week)
# groundwater <- groundwater %>%
#   group_by(well_id, year, isoweek) %>%
#   distinct(well_id, year, isoweek, .keep_all = TRUE) %>%
#   ungroup()

# get full range of year_weeks
year_range <- c(2018, 2019, 2021)
isoweek_range <- min(groundwater$isoweek):max(groundwater$isoweek)
year_week_range <- as.character(unlist(lapply(year_range, function(year) {
  paste0(year, sprintf("%02d", isoweek_range))
})))

# remove first two timesteps (201820, 201821 have no entries)
year_week_range <- year_week_range[-c(1,2)]
# remove last 2 weeks of 2018 and 2019, first 2 weeks of 2019
indices_to_remove <- c(21:26, 49, 50)
year_week_range <- year_week_range[-indices_to_remove]

# add new column with year and isoweek combined
# (e.g. 201820 for year 2018, week 20)
groundwater <- groundwater %>%
  mutate(year_week = as.character(paste0(year, sprintf("%02d", isoweek))))

# if multiple entries per well_id and year_week, average them
# --- TODO: is averaging the best representation of the data?
groundwater <- groundwater %>%
  group_by(well_id, year_week) %>%
  summarise(
    ground_to_water_cm = mean(ground_to_water_cm, na.rm = TRUE),
    .groups = "drop"
  )

# convert NaN to NA
groundwater$ground_to_water_cm[is.nan(groundwater$ground_to_water_cm)] <- NA

# compare original vs filtered entries (diff and percentage)
nrow_groundwater_orig - nrow(groundwater)
```

```
## [1] 365
```

```r
nrow(groundwater) / nrow_groundwater_orig * 100
```

```
## [1] 73.14202
```

```r
# completeness in terms of entries with NA (before filling in weeks)
groundwater %>%
  summarize(
    na_sum = sum(is.na(ground_to_water_cm)),
    na_percent = 100 * sum(is.na(ground_to_water_cm)) / n())
```

```
## # A tibble: 1 x 2
##   na_sum na_percent
##    <int>      <dbl>
## 1    135       13.6
```

```r
# create a complete grid off all well_id and year_week values
groundwater_full_grid <- expand_grid(
  well_id = unique(groundwater$well_id),
  year_week = year_week_range
)

# join the complete grid with groundwater
groundwater_full_grid <- groundwater_full_grid %>%
  left_join(groundwater, by = c("well_id", "year_week"))

# check for duplicates
duplicate_groundwater <- groundwater_full_grid %>%
  group_by(well_id, year_week) %>%
  summarize(
    count = n(),
    .groups = "drop"
  ) %>%
  filter(count>1)

# # create new dataframe with timesteps as columns and one unique well_id per row
groundwater_by_timestep <- groundwater_full_grid %>%
  pivot_wider(
    names_from = year_week,
    values_from = ground_to_water_cm,
    values_fill = NA
  )

# recheck completeness percentage
groundwater_by_timestep %>%
  summarize(
    na_sum = sum(across(everything(), ~ is.na(.))),
    na_percent = 100 * na_sum / (n() * ncol(.))
  )
```

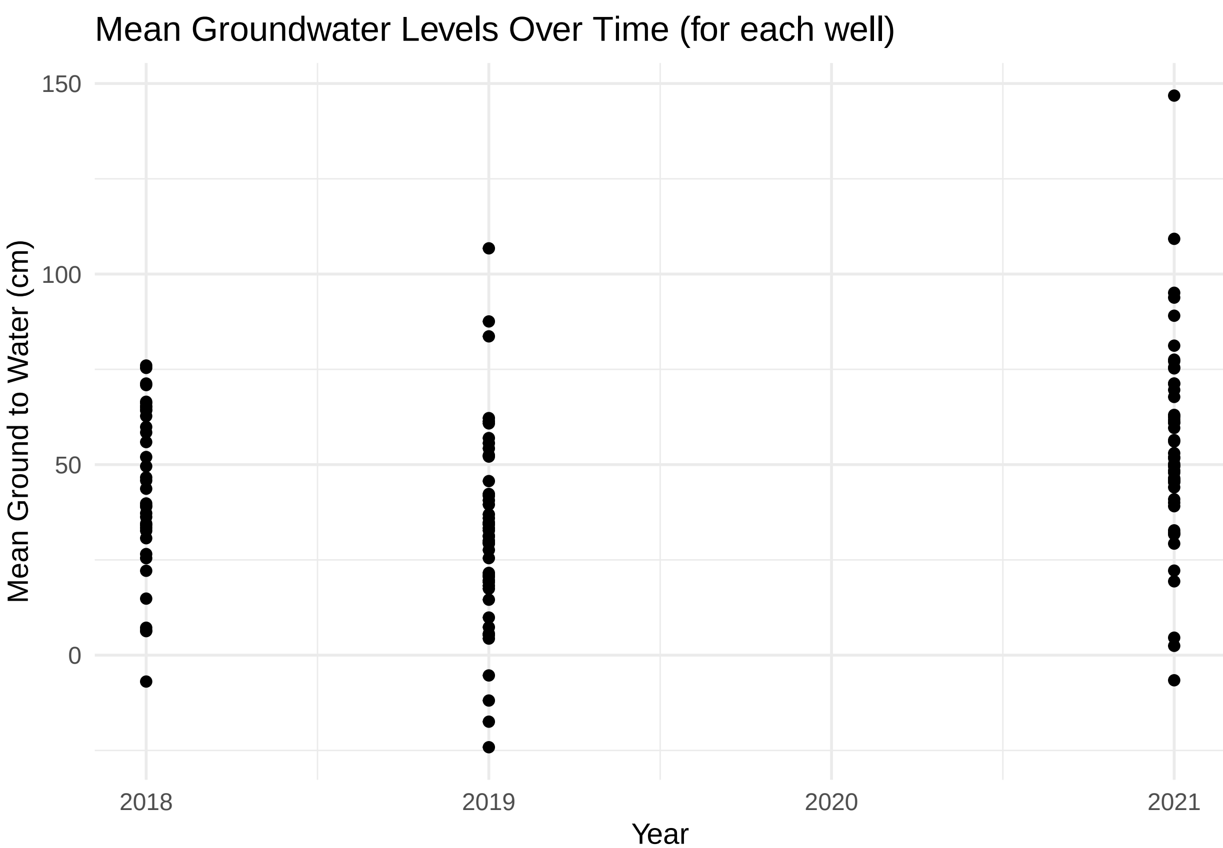
```
## # A tibble: 1 x 2
##   na_sum na_percent
##    <int>      <dbl>
## 1   2813       75.5
```

```r
# NOTE: 75.5% incomplete at weekly timestep with 2018-19 transducer data

# TODO Next step summary statistics
# ------wells in each meadow group: kiln, east, low
# ------wells from each plant functional type: sedge, willow, mixed herbaceous, pine
# ------wells from each hydrogeomorphic zone: riparian, terrace, fan

# TODO: Next time series to validate and prepare for analysis
# ------discharge (at one point)
# ------daily precipitation (at one point)
# ------sunlight, aka PAR (at one point)
# ------max, mean daily temperature (at each meadow)
```

Plots

1. Plot mean annual groundwater level for all wells for each year
2. Plot year-over-year daily time series of mean groundwater level

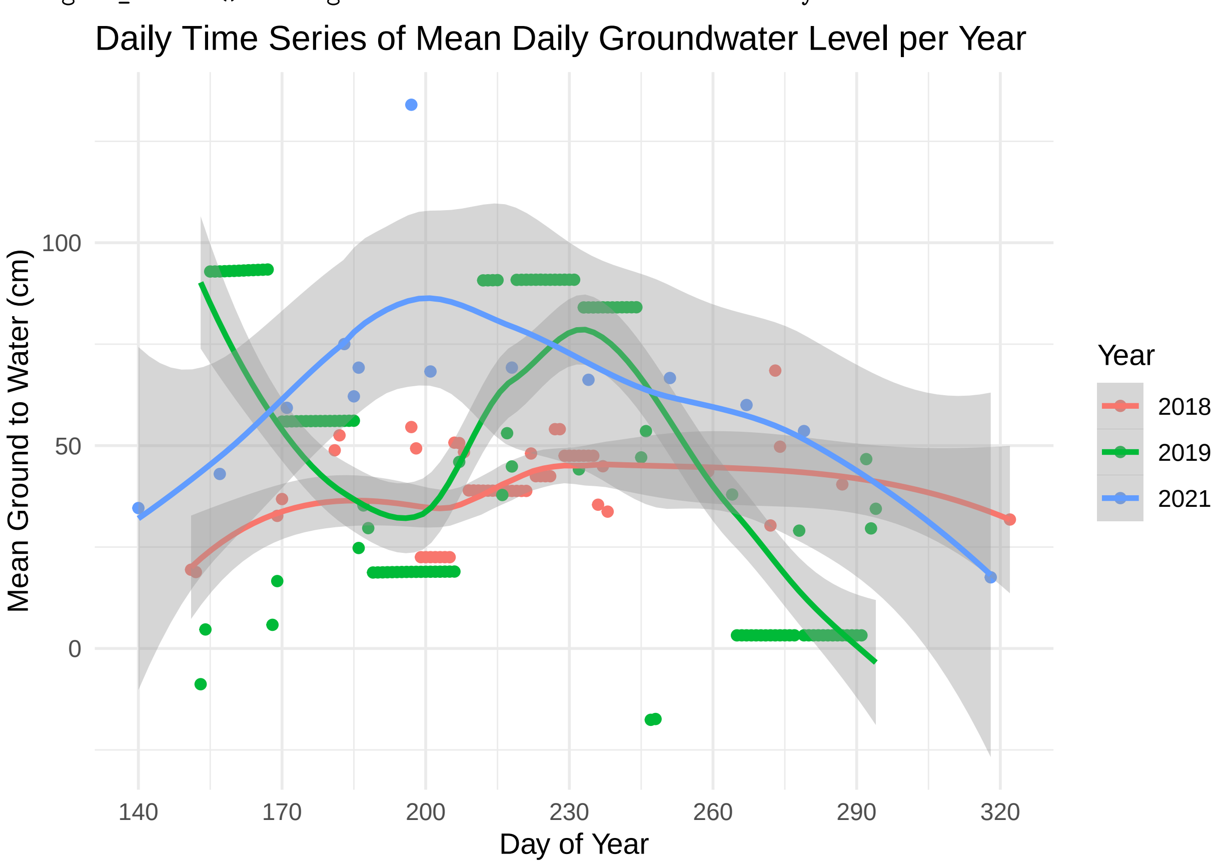
```r
# Load libraries
library(ggplot2)
library(tidyr)

# Plot (1): the mean annual groundwater level for all wells for each year
ggplot(groundwater_summary_by_well_year, aes(
  x = year,
  y = mean_value,
  group = well_id)) +
  geom_point() +  # Optional: add points at each data point
  labs(title = "Mean Groundwater Levels Over Time (for each well)",
       x = "Year", y = "Mean Ground to Water (cm)") +
  theme_minimal()
```




```r
# Plot (2): year-over-year daily time series of mean groundwater level

# ---Setup dataframe with new columns for year and day_of_year
groundwater_summary_by_day <- groundwater_summary_by_date %>%
  mutate(
    year = year(date),                 # Extract the year from Date
    day_of_year = yday(date)           # Extract the day of year (1-365/366)
  )
```

```r
# ---Add NA values for days with no measurement (or mean_value)
complete_groundwater_summary_by_day <- groundwater_summary_by_day %>%
  group_by(year) %>%
  complete(day_of_year =
           min(groundwater_summary_by_day$day_of_year):
             max(groundwater_summary_by_day$day_of_year),
           fill = list(mean_value = NA))   # Fill missing days with NA

# ---Plot it!
ggplot(complete_groundwater_summary_by_day, aes(
  x = day_of_year,
  y = mean_value,
  color = factor(year),
  group = year)) +
  geom_point() +
  geom_smooth() +
  theme_bw() +
  labs(title = "Daily Time Series of Mean Daily Groundwater Level per Year",
       x = "Day of Year",
       y = "Mean Ground to Water (cm)",
       color = "Year") +
  scale_x_continuous(breaks = seq(min(groundwater_summary_by_day$day_of_year),
                                  max(groundwater_summary_by_day$day_of_year),
                                  by = 30)) +  # Customize x-axis (days of year)
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Research Questions and Hypotheses

1. How does meadow groundwater vary by season and climate as influenced by elevation, hydrogeomorphic zones, and evapotranspiration rates of plant functional types?

- Hypothesis: I expect evapotranspiration to drive daily and seasonal groundwater levels with sensitivity to meteorology and day length.

2. What controls plant functional type phenology?

- Hypothesis: peak productivity and senescence will correlate to groundwater levels as governed by meteorology but moderated by hydrogeomorphic zones and elevation.

3. Does discharge, topography or subsurface character influence groundwater reliability?

- Hypothesis: I expect that groundwater reliability will correlate to topographic convergence or subsurface boundaries (i.e. differing conductivity).