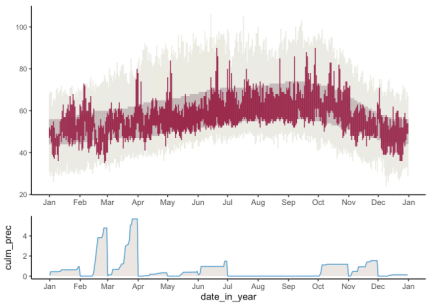
Adv Data Viz

```
library(dplyr)
library(tidyverse)
library(lubridate)
library(readr)

weather_clean <- weather_clean |>
  janitor::clean_names()

weather_clean$date_in_year <- as.Date(weather_clean$date_in_year - 1, or

temp <- ggplot(weather_clean, aes(x = date_in_year)) +
  geom_linerange(aes(
    ymin = record_low,
    ymax = record_high),
    color = "#ECEBE3") +
  geom_linerange(aes(
    ymin = normal_low,
    ymax = normal_high),
    color = "#C8B8BA") +
  geom_linerange(aes(
    ymin = low,
    ymax = high),
    color = "#A90248") +
  scale_x_date(
    date_labels = "%b",        # Short month names (Jan, Feb, ...)
    date_breaks = "1 month"    # Tick every month
  ) +
  theme_classic() +
  theme(axis.title = element_blank(),
    axis.text.y = element_text(size = 8),
    axis.text.x = element_text(size = 9),
    panel.grid = element_blank())

prec <- ggplot(weather_clean) +
  geom_area(aes(x = date_in_year, y = culm_prec), fill = "#ebeae2", colo
  geom_point(
    data = subset(weather_clean, record_precip == TRUE),
    aes(y = culm_prec),
    shape = 17,
    size = 2,
    color = "#32a3d8"
  ) +
  scale_x_date(date_labels = "%b", date_breaks = "1 month") +
  theme_classic()
```

```
library(patchwork)
```



Spatial Trimming Functions
st_bbox()
st_crop()

Colors
scale_color_viridis_c()
scale_fill_gradientn()

Saving a Plot
png("relative path to image", width = width_in_pixels, height = height_in_pixels)
# Code for creating plot
dev.off()

File Paths
- ../ parent directory
- ./ current directory

Logical Vectors
We can summarize logical vectors with:
any(): Are ANY of the values TRUE?
all(): Are ALL of the values TRUE?
sum(): How many of the values are TRUE?
mean(): What fraction of the values are TRUE?

Spatial Functions
st_crs()
st_transform(crs = "EPSG:32133")
mn_counties <- USAboundaries::us_counties(resolution = "high", states = "Minnesota")

elevation <- elevatr::get_elev_raster(mn_counties, z = 5, clip = "bbox")
raster::crs(elevation) <- sf::st_crs(mn_counties)

# Convert to data frame for plotting
elev_df <- elevation |> terra::as.data.frame(xy = TRUE)
colnames(elev_df) <- c("x", "y", "elevation")

6 Adv Data Wrangling P1

Logicals

```
x <- c(TRUE, FALSE, NA)
x
```

```
[1]  TRUE FALSE    NA
```

```
class(x)
```

```
[1] "logical"
```

You will often create logical vectors with comparison operators: >, <, <=, >=, ==, !=.

```
x <- c(1, 2, 9, 12)
x < 2
```

```
[1]  TRUE FALSE FALSE FALSE
```

```
x <= 2
```

```
[1]  TRUE  TRUE FALSE FALSE
```

```
x > 9
```

```
[1] FALSE FALSE FALSE  TRUE
```

```
x >= 9
```

```
[1] FALSE FALSE  TRUE  TRUE
```

```
x == 12
```

```
[1] FALSE FALSE FALSE  TRUE
```

```
x != 12
```

```
[1]  TRUE  TRUE  TRUE FALSE
```

When you want to check for set containment, the %in% operator is the correct way to do this (as opposed to ==).

```
x <- c(1, 2, 9, 4)
x == c(1, 2, 4)
```

```
Warning in x == c(1, 2, 4): longer object length is not a multiple of sho
object length

[1]  TRUE  TRUE FALSE FALSE
```

Adv Spatial Viz P1 CP

2. What are the two components of a CRS/GCS?
   Datum + Ellipsoid Model

3. Why is it insufficient to identify a location by its latitude and longitude?
   Longitude and latitude are calculated based on chosen a coordinate reference system (datum + ellipsoid model) and so that value of longitude and latitude of a location might change if the CRS changes.

4. Why do we need to be mindful about CRSs when working with different spatial datasets?

- Different CRSs may use different units (e.g., meters, feet). Consistency in units is essential for accurate distance and area calculations.
- Joining datasets from multiple sources requires a common CRS.
- Different datasets may use different CRSs. If these CRSs are not accounted for, the spatial points/regions will not align correctly on a map.

Adv Data Wrangling P2 CP

3. The correct way to recreate paste0("Letter of the Day:",letters) with str_c() is
   str_c("Letter of the Day:",letters, sep = '')

4. "a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z"
   str_c(letters, collapse = ',')
   str_flatten(letters,',')

5. To separate a vector of strings (deliminted by commas) into columns of strings, use
   separate_wider_delim()

6. For the following string,

x1 <- "textNi1o was particularly bad this year"

one of these functions will give you meaningful data and one will not. Why?

read_csv(x1)$text

read_csv(x1, locale = locale(encoding = "Latin1"))$text
Non-English (non-ASCII) characters can be encoded in a wide variety of ways and read_csv() uses UTF-8 encoding by default.

7. Which of these characters means that a pattern is optional (matches 0 or 1 time)?
?

8. Which of these characters means that a pattern is repeats (at least once)?
+

7 Adv Data Wrangling P2

```
library(stringr)
string1 <- "This is a string"
string2 <- 'If I want to include a "quote" inside a string, I use single
string3 <- c(string1, string2) # string / character vector (of greater t

str_view(string1, html = TRUE)
```

```
This is a string
```

Creates a tab) (Creates a newline)

Creating Strings

```
df_dates <- tibble(
    year = c(2000, 2001, 2002),
    month = c("Jan", "Feb", "Mar"),
    day = c(3, 4, 5)
)

df_dates |>
    mutate(
        date1 = str_c(month, "-", day, "-", year),
        date2 = str_glue("{month}-{day}-{year}")
    )
```

```
# A tibble: 3 × 5
   year month   day date1      date2
  <dbl> <chr> <dbl> <chr>      <glue>
1  2000 Jan       3 Jan-3-2000 Jan-3-2000
2  2001 Feb       4 Feb-4-2001 Feb-4-2001
3  2002 Mar       5 Mar-5-2002 Mar-5-2002
```

Extracting Information with String

```
df <- tibble(
    word_id = 1:3,
    word = c("replace", "match", "pattern")
)

df |>
    mutate(
        word_length = str_length(word),
        middle_pos = ceiling(word_length/2),
        middle_letter = str_sub(word, middle_pos, middle_pos)
    )
```

```
# A tibble: 3 × 5
  word_id word    word_length middle_pos middle_letter
    <int> <chr>         <int>      <dbl> <chr>
1       1 replace           7          4 l
2       2 match             5          3 t
3       3 pattern           7          4 t
```