

# climateAnalyzeR Vignette

Matt Van Soyoc

This package imports annual, monthly, and daily weather data from The Climate Analyzer ([ClimateAnalyzer.org](http://ClimateAnalyzer.org)) into R. This vignette demonstrates how to use the various functions in the `climateAnalyzeR` package to import and visualize temperature and precipitation data from [Global Historical Climatology Network \(GHCN\)](#) Co-op weather stations.

This package can also produce brief water year (Oct-Sep) and annual (Jan-Dec) climate reports using temperature and precipitation data from GHCN Co-op stations using RMarkdown. However, this function is currently broken and I'm working on fixing it. I will include an example here once it's fixed.

## Download pacakge

This package is currently hosted on GitHub. You can use the `devtools` or `remotes` packages to download `climateAnalyzeR` into your library. In this example, we are going to use `devtools`. Let's first install and load the packages used in this vignette, then we'll install `climateAnalyzeR` from GitHub.

```
# List of packages used in this vignette
pkgs <- c('arcgislayers', # reading feature layers from ArcGIS Online
         'cowplot',      # arranging multiple ggplots
         'devtools',     # installing packages from GitHub
         'dplyr',        # data wrangling
         'ggplot2',      # creating figures
         'janitor',     # cleaning data
         'maptiles',     # getting base map tiles
         'sf',           # working with spatial data
         'tmap')         # mapping

# Install packages if it's not already in your library
inst_pkgs <- pkgs %in% rownames(installed.packages())
if (any(inst_pkgs == FALSE)) {
```

```

install.packages(pkgs[!inst_pkgs],
                 lib = .libPaths()[1],
                 repos = "https://cloud.r-project.org",
                 type = 'source',
                 dependencies = TRUE,
                 quiet = TRUE)
}

# Load packages
invisible(lapply(pkgs, library, character.only = TRUE))

# Install climateAnalyzeR from GitHub
devtools::install_github("scoyoc/climateAnalyzeR")
#>
#> -- R CMD build -----
#> * checking for file 'C:\Users\scoyo\AppData\Local\Temp\RtmpyKD7vV\remotesb59c40d27a\sc
#> * preparing 'climateAnalyzeR':
#> * checking DESCRIPTION meta-information ... OK
#> * checking for LF line-endings in source and make files and shell scripts
#> * checking for empty or unneeded directories
#>   NB: this package now depends on R (>= 4.1.0)
#>   WARNING: Added dependency on R >= 4.1.0 because package code uses the
#>   pipe |> or function shorthand \(...) syntax added in R 4.1.0.
#>   File(s) using such syntax:
#>     'annual_figure.Rd' 'append_water_yr_mth.R' 'append_water_yr_mth.Rd'
#>     'departure_figure.Rd' 'import_departure.R' 'import_monthly.R'
#>     'monthly_figure.Rd' 'normals.R' 'renderSummaries.R' 'stations.R'
#>     'utils.R'
#> * building 'climateAnalyzeR_0.9.2.tar.gz'
#>

# Load climateAnalyzeR
library('climateAnalyzeR')

```

## Read data into R

Several functions read data from The Climate Analyzer into R. Below are examples of how to use these functions to import annual, monthly, and daily temperature and precipitation data.

## GHCN Co-op weather stations

First, we need to get a list of GHCN Co-op stations hosted on The Climate Analyzer. The `stations()` function retrieves these data. In this example we are going to use the `type = "COOP"` argument to get a list of Co-op stations.

```
ghcn_stations <- stations(type = "COOP")

dplyr::glimpse(ghcn_stations)
#> #> Rows: 656
#> #> Columns: 9
#> #> $ station_name    <chr> "Aberdeen Experimental Station", "Alta 1NNW", "Antelope-
#> #> $ id              <chr> "USC00100010", "480140", "350197", "470349", "100470", ~
#> #> $ type             <chr> "COOP", "COOP", "COOP", "COOP", "COOP", "COOP", "COOP", ~
#> #> $ lat              <dbl> 42.95300, 43.77270, 44.81972, 46.56660, 44.04240, 32.33~
#> #> $ lon              <dbl> -112.8250, -111.0340, -120.7533, -90.9660, -111.2740, ~
#> #> $ elev_m            <dbl> 1342, 1962, 924, 198, 1589, 1151, 305, 200, 200, 1263, ~
#> #> $ years_avail       <chr> "1914;1915;1916;1917;1918;1919;1920;1921;1922;1923;1924~
#> #> $ years_segments    <chr> "1914 - 2025", "1909 - 1940;1948 - 2025", "1924 - 1943;~
#> #> $ data_url           <chr> "ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/all/USC001~
```

There are 656 GHCN Co-op station hosted on The Climate Analyzer. Lets filter this down to a more focused area, like Canyonlands National Park (NP), Utah.

For this example we are going to use the `arcgislayers` package to read in the National Park Service (NPS) park boundaries from ArcGIS Online. The URL below is a publicly available REST endpoint hosted by the National Park Service. These data need to be transformed to EPSG:4326 (WGS 84) so they align with the GHCN Co-op station coordinates. Next, we are going to use the `sf` package to clip the GHCN Co-op stations to Canyonlands NP. Finally, we will include the Hans Flat Ranger Station Co-op station just outside the park boundary.

```
# URL for NPS boundaries on ArcGIS Online
rest_url = "https://services1.arcgis.com/fBc8EJBxQRMcHlei/arcgis/rest/services/NPS_Land_Re

# Read feature layer and filter to Canyonlands NP
cany <- arcgislayers::arc_read(rest_url) |>
  janitor::clean_names() |>
  sf::st_make_valid() |>
  dplyr::filter(unit_code == "CANY") |>
  sf::st_transform(crs = "EPSG:4326") # Transform data to WGS 84

# Convert to sf object
```

```

ghcn_sf <- sf::st_as_sf(ghcn_stations, coords = c("lon", "lat"),
                        crs = "EPSG:4326") |>
  sf::st_make_valid()

# Clip GHCN Co-op stations in Canyonlands NP
cany_ghcn <- sf::st_intersection(ghcn_sf, cany) |>
  dplyr::select(-tidyselect::any_of(colnames(cany))) |>
  # Include the Hans Flat RS station just outside the park boundary
  dplyr::bind_rows(
    dplyr::filter(ghcn_sf, station_name == "Hans Flat RS")
  )

# View data
dplyr::glimpse(cany_ghcn)
#> Rows: 5
#> Columns: 8
#> $ station_name      <chr> "Canyonlands The Neck", "Canyonlands The Needle", "Gray~
#> $ id                 <chr> "421163", "421168", "XXX3", "XXX4", "423600"
#> $ type               <chr> "COOP", "COOP", "COOP", "COOP", "COOP"
#> $ elev_m              <dbl> 1790, 1495, 0, 0, 2012
#> $ years_avail        <chr> "1965;1966;1967;1968;1969;1970;1971;1972;1973;1974;1975~
#> $ years_segments     <chr> "1965 - 2025", "1965 - 2025", "1980 - 2025", "1980 - 20~
#> $ data_url            <chr> "ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/all/USC004~
#> $ geometry            <POINT [°]> POINT (-109.821 38.46), POINT (-109.759 38.167), POINT ~

```

Now let's make a quick map using the `maptiles` and `tmap` packages to visualize the GHCN Co-op stations in Canyonlands National Park.

```

# Get base tiles
aoa_tiles <- maptiles::get_tiles(cany, provider = "Esri.WorldTerrain",
                                    zoom = 12)
#>
|-----|-----|-----|-----|
=====

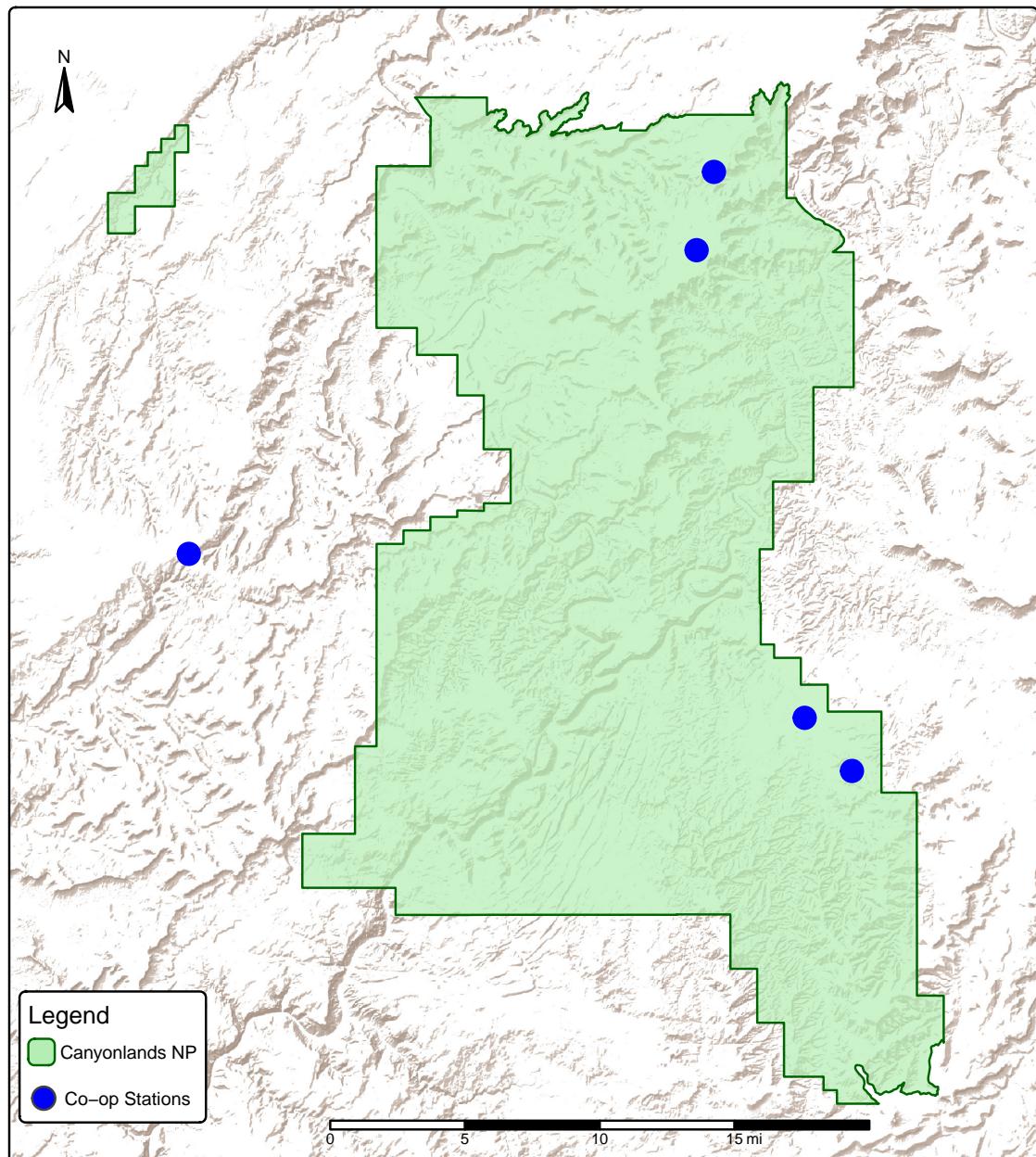
|-----|-----|-----|-----|
=====

# Map
tmap::tm_shape(aoa_tiles, unit = "mi") + tmap::tm_rgb() +

```

```
tmap::tm_shape(cany) + tmap::tm_polygons(fill = 'darkseagreen2',
                                         col = 'darkgreen',
                                         fill_alpha = 0.7) +
tmap::tm_shape(cany_ghcn) + tmap::tm_symbols(shape = 20, fill = 'blue') +
tmap::tm_compass(type = "arrow", position = c("left", "top"),
                  text.size = 0.5, bg.color = 'white', bg.alpha = 0.5) +
tmap::tm_scalebar(breaks = c(0, 5, 10, 15, 20), position = c(0.3, 0.05),
                  text.size = 0.4, bg.color = 'white', bg.alpha = 0.5) +
tmap::tm_add_legend(type = "polygons", labels = "Canyonlands NP",
                     fill = "darkseagreen2", col = 'darkgreen',
                     title = "Legend") +
tmap::tm_add_legend(type = "symbols", labels = "Co-op Stations",
                     fill = "blue", size = 0.7) +
tmap::tm_layout(main.title.size = 1, legend.position = c(0.01, 0.15),
               legend.text.size = 0.5, legend.title.size = 0.7,
               legend.frame = TRUE, legend.bg.color = 'white',
               frame = TRUE) +
tmap::tm_title("Canyonlands NP GHCN Co-op Stations")
```

## Canyonlands NP GHCN Co-op Stations



## Finding your station\_id

Unfortunately, there isn't a good way to read the `station_id` into R. Sorry about that! :/

To get the `station_id`, go to [The Climate Analyzer](https://www.climateanalyzer.net/coop) and select the station you want to download data for and get the station id from the URL. For this vignette we are going to pull data from the GHCN Co-op station at the Island in the Sky Visitor Center in Canyonlands National Park . This station is named “Canyonlands The Neck” on The Climate Analyzer. The `station_id` is the last section of the URL: `canyonlands_theneck`.

The screenshot shows a web browser window with the URL `https://www.climateanalyzer.net/coop/canyonlands_theneck` in the address bar. The page title is "Canyonlands The Neck". Below the title, there is a link "Show Station Information". A question "Do you want graphs or tables for this station?" is followed by a dropdown menu set to "Graphs or Tables". The top navigation bar includes links for Home, Yellowstone and Grand Teton, Bighorn Canyon, Mediterranean Coast, Sonoran Desert, Great Lakes, Northern Colorado Plateau, Mojave Desert, Rocky Mountain, Southern Colorado Plateau, Chihuahuan Desert, and North Coast and Cascades.

## Importing normalized temperature and precipitation data

The `normals()` function imports NOAA calculated temperature and precipitation normals for a specified reference period. In this example, we are going to use the 1981-2010 reference period. Other options available are 1971-2000 and 1991-2020.

```
isky_normals <- normals(station_id = "canyonlands_theneck",
                           ref_period = "1981-2010")

dplyr::glimpse(isky_normals)
#> Rows: 39
#> Columns: 5
#> $ station_id <chr> "canyonlands_theneck", "canyonlands_theneck", "canyonlands_~
#> $ id          <chr> "421163", "421163", "421163", "421163", "421163", "421163", ~
#> $ element     <chr> "PRCP", "PRCP", "PRCP", "PRCP", "PRCP", "PRCP", "PR~
#> $ month       <fct> Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec, ~
#> $ value        <dbl> 0.64, 0.55, 0.79, 0.71, 0.70, 0.44, 0.89, 1.11, 1.09, 1.30, ~
```

## Import annual temperature and precipitation data

The `import_annual()` function imports annual temperature and precipitation data for a specified station and time period. In this example we are going to import all the data (1965 to 2025) for the Island in the Sky Co-op station.

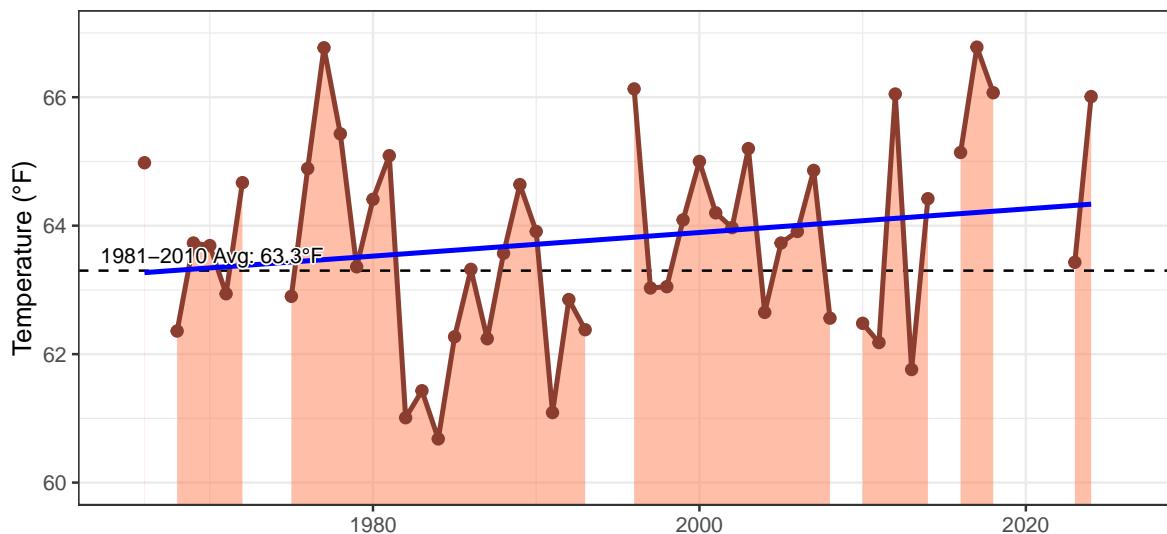
```
isky_annual <- import_annual("canyonlands_theneck", 1965, 2025)

dplyr::glimpse(isky_annual)
#> #> Rows: 62
#> #> Columns: 7
#> #> $ year      <dbl> 1965, 1966, 1967, 1968, 1969, 1970, 1971, 1972, 1973, 1974, 19~
#> #> $ prcp_cy <dbl> NA, 8.76, 8.66, 6.03, 12.38, 7.22, 7.45, 9.61, NA, NA, 10.07, ~
#> #> $ prcp_wy <dbl> NA, 10.66, 8.54, 7.08, 11.96, 7.37, 5.22, 6.39, 12.94, NA, 11.~
#> #> $ tmax_cy <dbl> NA, 64.98, NA, 62.36, 63.73, 63.69, 62.94, 64.67, NA, NA, 62.9~
#> #> $ tmax_wy <dbl> NA, NA, 63.91, NA, 64.04, 63.25, 63.77, 64.99, 60.60, NA, 61.7~
#> #> $ tmin_cy <dbl> NA, 41.87, 41.97, 40.61, 42.55, 42.25, 41.60, 43.29, NA, NA, 4~
#> #> $ tmin_wy <dbl> NA, 42.50, 41.94, 41.15, 42.47, 41.88, 42.28, 43.28, 39.34, NA~
```

Now let's use the `annual_figure()` function to plot annual maximum temperature (TMAX), minimum temperature (TMIN), and precipitation (PRCP) data along with the 1981-2010 normals.

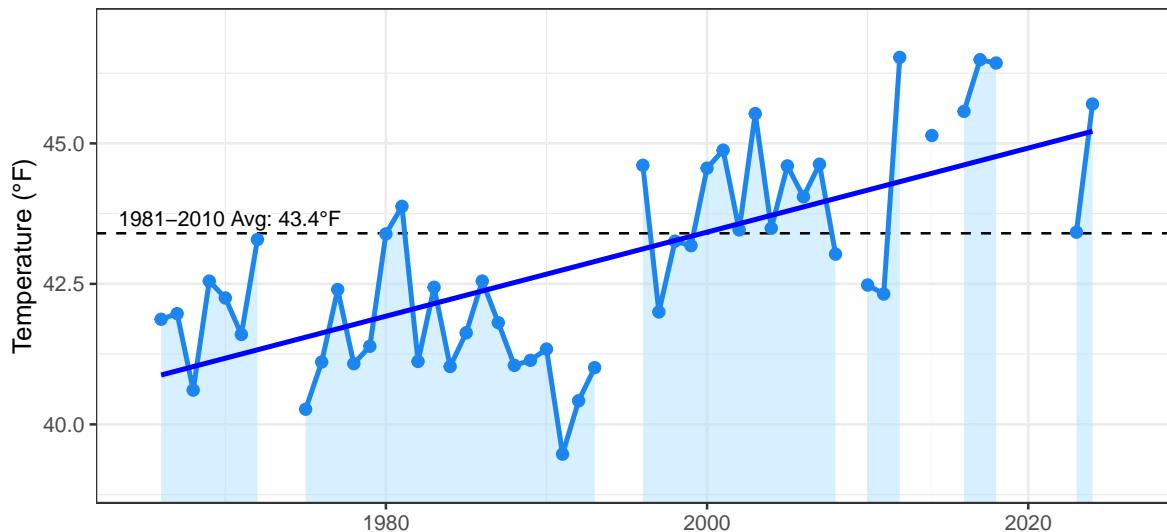
```
# Maximum Temperature (TMAX) ----
# Get annual maximum temperature normals
ann_tmax_normals <- dplyr::filter(isky_normals,
                                    element == "TMAX" & month == "Annual")
# Plot annual maximum temperature data
my_title <- "Annual Maximum Temperature (1965-2025)"
annual_figure(x_var = isky_annual$year, y_var = isky_annual$tmax_cy,
              normal = ann_tmax_normals$value, reference_period = "1981-2010",
              area_color = "coral", line_color = "coral4",
              my_title = my_title, my_ylab = "Temperature (°F)")
```

### Annual Maximum Temperature (1965–2025)

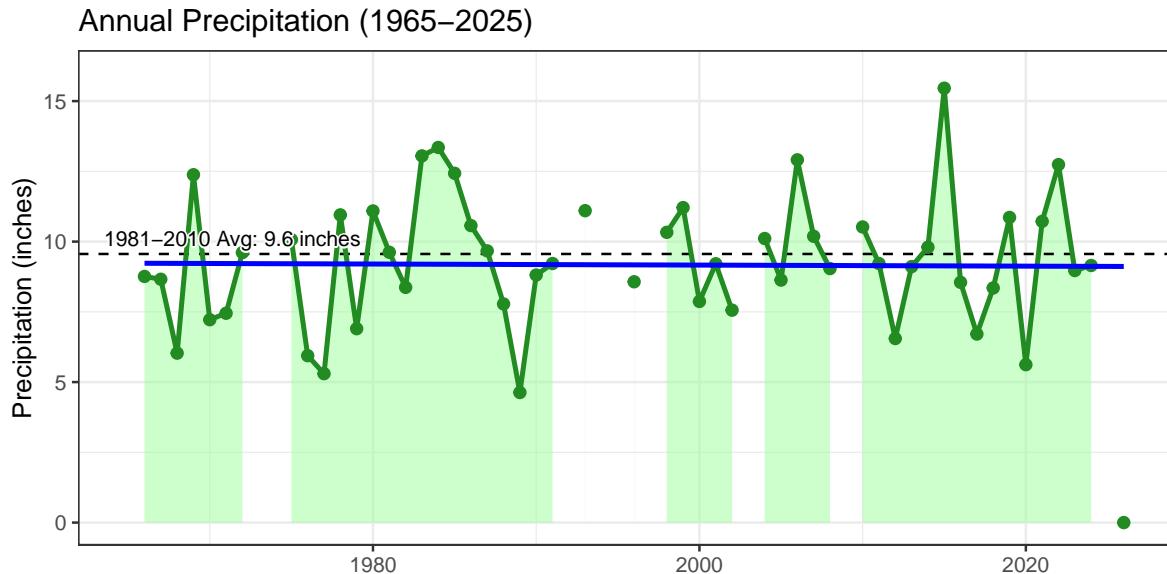


```
# Minimum Temperature (TMIN) ----
# Get annual minimum temperature normals
ann_tmin_normals <- dplyr::filter(isky_normals,
                                    element == "TMIN" & month == "Annual")
# Plot annual minimum temperature data
my_title <- "Annual Minimum Temperature (1965-2025)"
annual_figure(x_var = isky_annual$year, y_var = isky_annual$tmin_cy,
              normal = ann_tmin_normals$value, reference_period = "1981-2010",
              area_color = "lightskyblue1", line_color = "dodgerblue2",
              my_title = my_title, my_ylab = "Temperature (°F)")
```

### Annual Minimum Temperature (1965–2025)



```
# Precipitation (PRCP) ----
# Get annual precipitation normals
ann_prcp_normals <- dplyr::filter(isky_normals,
                                    element == "PRCP" & month == "Annual")
# Plot annual precipitation data
my_title <- "Annual Precipitation (1965–2025)"
annual_figure(x_var = isky_annual$year, y_var = isky_annual$prcp_cy,
              normal = ann_prcp_normals$value, reference_period = "1981–2010",
              area_color = "palegreen1", line_color = "forestgreen",
              my_title = my_title, my_ylab = "Precipitation (inches)")
```



## Import monthly temperature and precipitation data

Next, we are going to use the `import_monthly()` function to import monthly temperature and precipitation data for the Island in the Sky Co-op station from 1965 to 2025. After importing the data, we are going to create a water year (Oct-Sep) variable and a factor variable for month names so we can plot the data by water year using the `append_water_yr_mth()` function.

```
isky_monthly <- import_monthly("canyonlands_theneck", 1965, 2025) |>
  append_water_yr_mth()

dplyr::glimpse(isky_monthly)
#> Rows: 732
#> Columns: 10
#> $ year      <dbl> 1965, 1965, 1965, 1965, 1965, 1965, 1965, 1965, 1965~  

#> $ month     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7~  

#> $ prcp       <dbl> NA, NA, NA, NA, NA, 2.21, 0.68, 0.70, NA, NA, 0.91, 0.~  

#> $ tmax       <dbl> NA, NA, NA, NA, NA, 90.61, 89.00, 74.18, NA, NA, 39.00~  

#> $ tmin       <dbl> NA, NA, NA, NA, NA, 63.36, 61.04, 49.54, NA, NA, 23.84~  

#> $ water_month <int> 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1~  

#> $ month_name <fct> Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec~  

#> $ season     <chr> "Wi", "Wi", "Sp", "Sp", "Sp", "Su", "Su", "Su", "Fa", "Fa"~  

#> $ water_yr    <dbl> 1965, 1965, 1965, 1965, 1965, 1965, 1965, 1965, 1966~  

#> $ water_mth   <fct> Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec~
```

We can plot monthly maximum temperature (TMAX), minimum temperature (TMIN), and precipitation (PRCP) data using the `monthly_figure()` function. We are going to use the `cowplot` package to plot the three figures in a single column in one figure.

```
plot_title <- cowplot::ggdraw() +
  cowplot::draw_label("Water Year Monthly Climate Data (1965-2025)",
                      fontface = 'bold', size = 16, x = 0.5, y = 0.5)

# Monthly TMAX figure
mth_tmax <- monthly_figure(
  x_var = isky_monthly$water_mth, y_var = isky_monthly$tmax,
  year_group = isky_monthly$water_yr, my_year = 2025, line_color = "coral4",
  my_title = "A. Monthly Maximum Temperature",
  my_ylab = "Temperature (°F)"
)

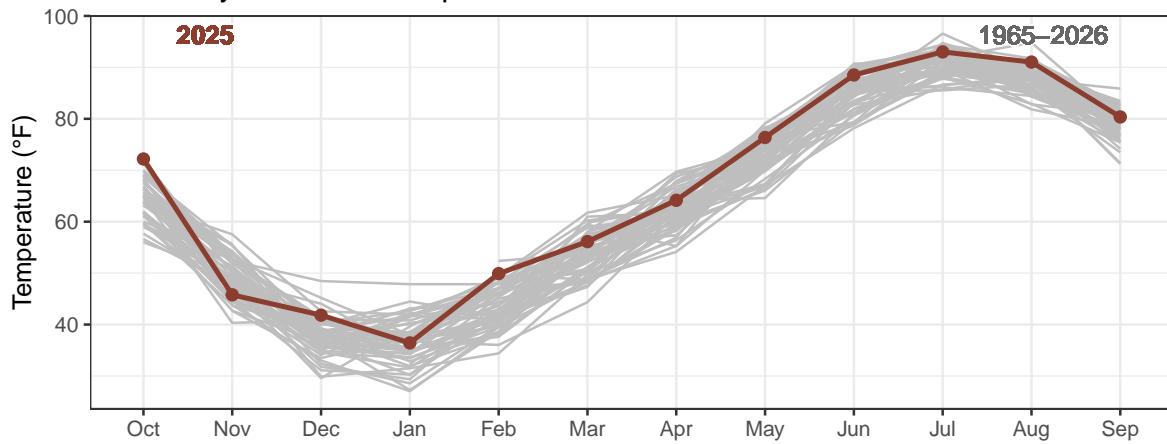
# Monthly TMIN figure
mth_tmin <- monthly_figure(
  x_var = isky_monthly$water_mth, y_var = isky_monthly$tmin,
  year_group = isky_monthly$water_yr, my_year = 2025, line_color = "dodgerblue2",
  my_title = "B. Monthly Minimum Temperature",
  my_ylab = "Temperature (°F)"
)

# Monthly PRCP figure
mth_prcp <- monthly_figure(
  x_var = isky_monthly$water_mth, y_var = isky_monthly$prcp,
  year_group = isky_monthly$water_yr, my_year = 2025, line_color = "forestgreen",
  my_title = "C. Monthly Precipitation",
  my_ylab = "Precipitation (inches)"
)

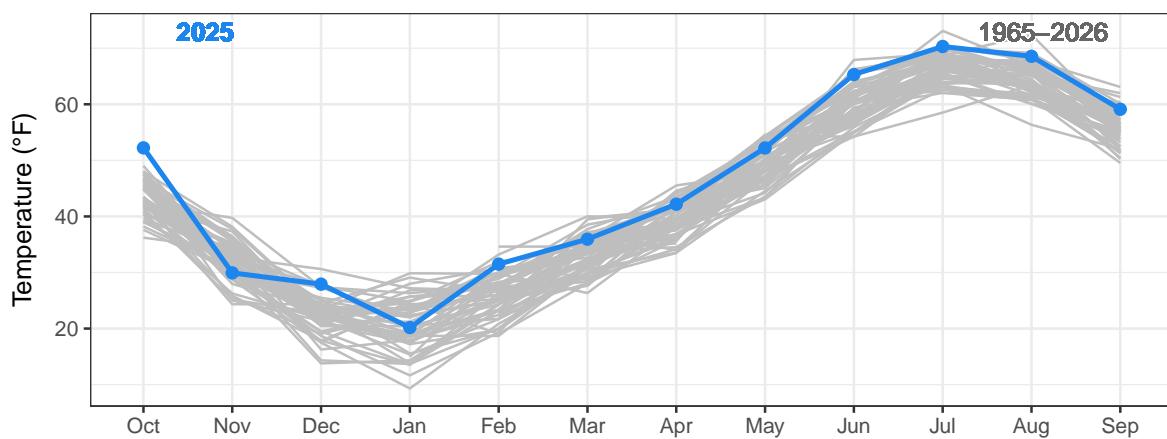
# Combine figures
cowplot::plot_grid(plot_title, mth_tmax, mth_tmin, mth_prcp,
                    labels = c(), ncol = 1, align = "v", vjust = 1, scale = 1,
                    rel_heights = c(0.1, 1, 1, 1))
```

## Water Year Monthly Climate Data (1965–2025)

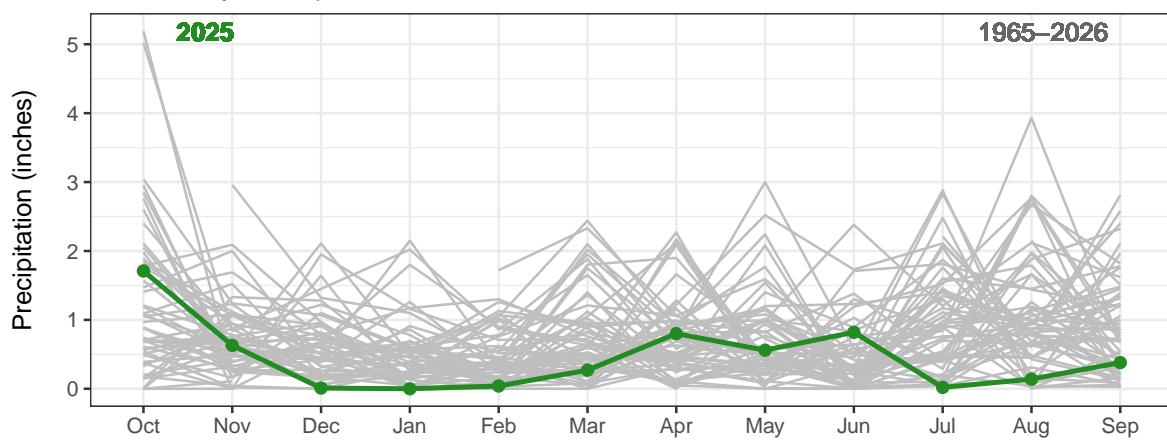
A. Monthly Maximum Temperature



B. Monthly Minimum Temperature



C. Monthly Precipitation



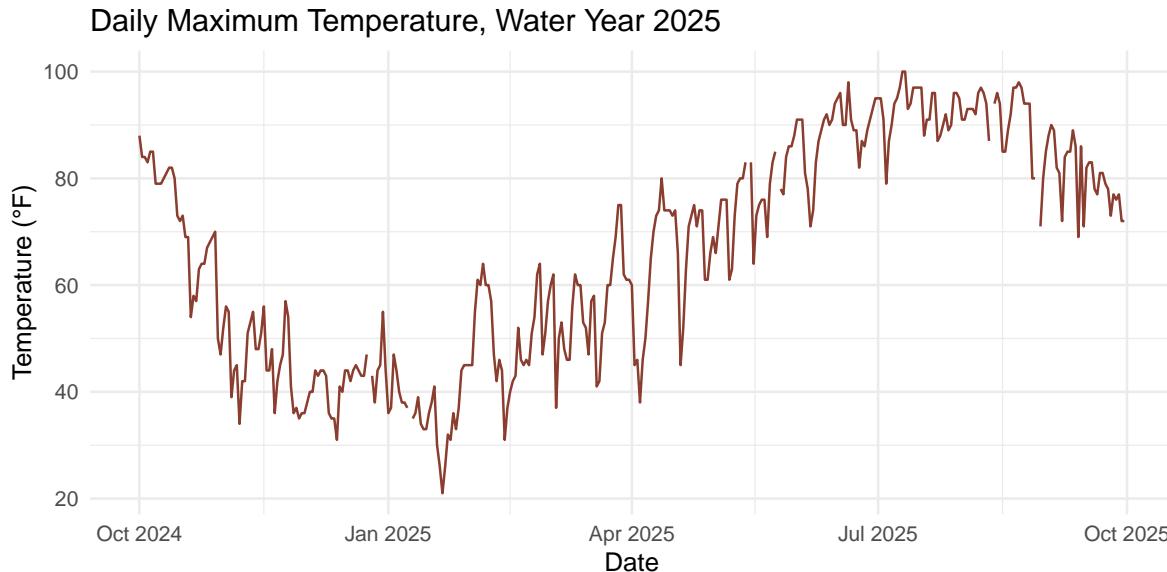
## Import daily temperature and precipitation data

Next, we are going to use the `import_daily()` function to import daily temperature and precipitation data for the Island in the Sky Co-op station from 1964 to 2025. Again, we are going to add the water year (Oct-Sep) and water month variables using the `append_water_yr_mth()` function.

I haven't used the daily data much yet, so I don't have any plotting functions for daily data. However, you can use `ggplot2` to create your own plots. Here's a quick example of how to plot daily maximum temperature (TMAX) data for water year 2025.

```
# Filter data for water year 2025
isky_daily_2025 <- dplyr::filter(isky_daily, water_yr == 2025)

# Plot daily maximum temperature data
ggplot2::ggplot(data = isky_daily_2025, aes(x = date, y = tmax_f)) +
  ggplot2::geom_line(color = "coral4") +
  ggplot2::labs(title = "Daily Maximum Temperature, Water Year 2025",
                x = "Date", y = "Temperature (°F)") +
  ggplot2::theme_minimal()
```



## Import departure data

The `import_departure()` function imports monthly temperature and precipitation departure data for a specified station and time period. In this example we are going to import monthly departure data for the Island in the Sky Co-op station from 1965 to 2025 and add the water year (Oct-Sep) and water month variables using the `append_water_yr_mth()` function..

```
isky_depart <- import_departure("canyonlands_theneck", 1965, 2025) |>
  append_water_yr_mth()

dplyr::glimpse(isky_depart)
#> #> Rows: 732
#> #> Columns: 10
#> #> $ year      <dbl> 1965, 1965, 1965, 1965, 1965, 1965, 1965, 1965, 1965~
#> #> $ month     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7~
#> #> $ prcp_pctavg <dbl> NA, NA, NA, NA, NA, 228.10, 87.94, 80.08, NA, NA, 166.~
#> #> $ tmax_depart <dbl> NA, NA, NA, NA, NA, 0.11, 1.00, -4.32, NA, NA, 1.20, ~~
#> #> $ tmin_depart <dbl> NA, NA, NA, NA, NA, -1.74, -2.26, -5.16, NA, NA, 2.84, ~
#> #> $ water_month <int> 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1~
#> #> $ month_name <fct> Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec~
#> #> $ season      <chr> "Wi", "Wi", "Sp", "Sp", "Sp", "Su", "Su", "Su", "Fa", "Fa"~
#> #> $ water_yr     <dbl> 1965, 1965, 1965, 1965, 1965, 1965, 1965, 1965, 1965, 1966~
#> #> $ water_mth    <fct> Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec~
```

Now we can plot monthly temperature (TMAX & TMIN) and precipitation (PRCP) departure data using the `departure_figure()` function.

```
plot_title <- cowplot::ggdraw() +
  cowplot::draw_label("Water Year Monthly Climate Departures (1965–2025)",
                      fontface = 'bold', size = 16, x = 0.5, y = 0.5)

# Departure TMAX figure
depart_tmax <- departure_figure(
  x_var = isky_depart$water_mth, y_var = isky_depart$tmax_depart,
  year_group = isky_depart$water_yr, y_intercept = 0, my_year = 2025,
  line_color = "coral4",
  my_title = "A. Monthly Maximum Temperature Departure",
  my_ylab = "Departure (°F)"
)

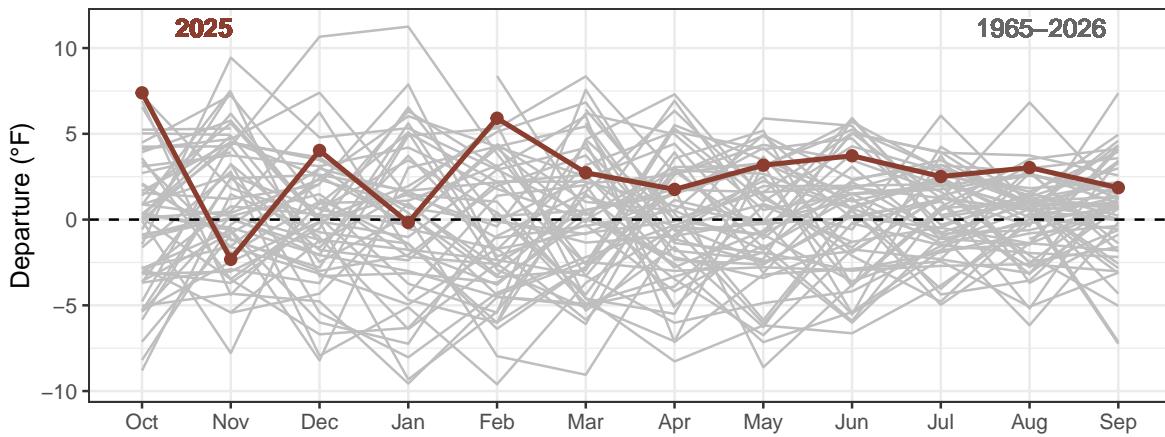
# Departure TMIN figure
depart_tmin <- departure_figure(
  x_var = isky_depart$water_mth, y_var = isky_depart$tmin_depart,
  year_group = isky_depart$water_yr, y_intercept = 0, my_year = 2025,
  line_color = "dodgerblue2",
  my_title = "B. Monthly Minimum Temperature Departure",
  my_ylab = "Departure (°F)"
)

# Departure PRCP figure
depart_prcp <- departure_figure(
  x_var = isky_depart$water_mth, y_var = isky_depart$prcp_pctavg,
  year_group = isky_depart$water_yr, y_intercept = 100, my_year = 2025,
  line_color = "forestgreen",
  my_title = "C. Monthly Precipitation Departure",
  my_ylab = "Departure (%)"
)

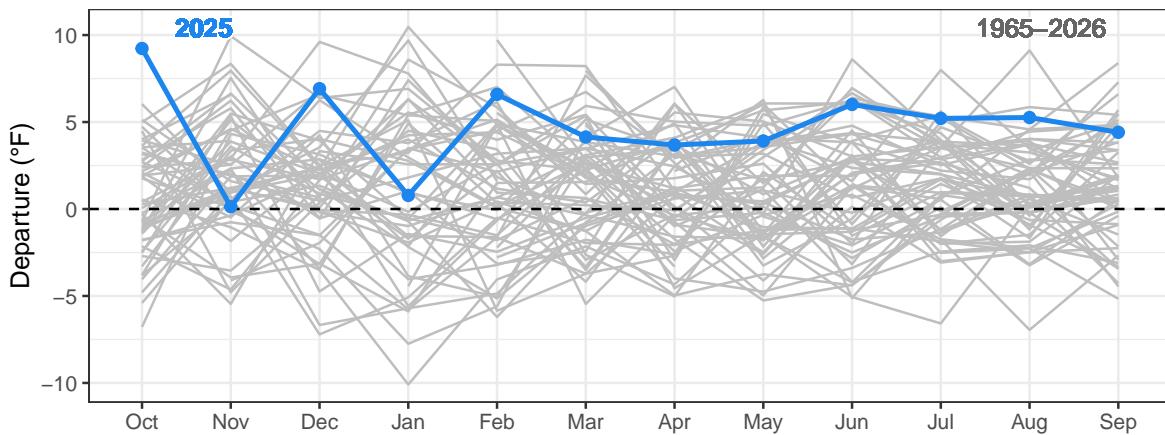
# Combine figures
cowplot::plot_grid(plot_title, depart_tmax, depart_tmin, depart_prcp,
                    labels = c(), ncol = 1, align = "v", vjust = 1, scale = 1,
                    rel_heights = c(0.1, 1, 1, 1))
```

## Water Year Monthly Climate Departures (1965–2025)

A. Monthly Maximum Temperature Departure



B. Monthly Minimum Temperature Departure



C. Monthly Precipitation Departure

