Creating Montana's Climate Normals from 1991 to 2020

This is an overview on how to use the `normals` package to create climate normals for the state of Montana. However, this process will work for any region within the contiguous U.S. that overlaps with the gridMET data we are using to create normals. First, we will load the `normals` and `magrittr` packages:

```r
library(normals)
library(magrittr)

# You can change this directory as needed.
data_dir = "~/data/mt_normals"
# You can specify more or fewer variables here. See ?normals::fetch_gridmet
# for other options.
variables = c("pr", "tmmn", "tmmx")
```

The next step is to download gridMET data to perform the analysis. Here we are going to specify a `data_dir` directory that will be used for the entire analysis. This directory will have several sub-directories that store data at different stages of analysis. We recommend using a new directory without any existing data to keep things tidy. The `normals::fetch_gridmet` function will create the specified directory and save data to `your-data-dir/raw`.

```r
normals::fetch_gridmet(
  data_dir = data_dir,
  variables = variables,
  start_year = 1991,
  end_year = 2020
)
```

The next important step is to crop the gridMET data to our area of interest (the state of Montana in our case). There is a helper function in the `normals` package called `crop_to_roi` that will crop all the data we just downloaded and save it out as a new file.

```r
# Choose a new directory to save the data for Montana out to.
mt_dir <- file.path(data_dir, "processed", "montana")

normals::crop_to_roi(
  # The previous function saved data to the `raw` subdirectory.
  data_dir = file.path(data_dir, "raw"),
  out_dir = mt_dir,
  roi = normals::mt,
  variables = variables
)
```

The data that we just downloaded and cropped are provided at a daily timestep, but we want to calculate monthly and annual normals for all of our data. To simplify this process, we will now aggregate the daily data to monthly and annual timescales using the `normals::aggregate_daily` function:

```r
out_dir <- file.path(data_dir, "processed", "montana", "summarized")
if (!dir.exists(out_dir)) {
  dir.create(out_dir)
}
```

```r
lapply(variables, function(x) {
  r <- list.files(mt_dir, recursive = T, pattern = x, full.names = T) %>%
    terra::rast()

  # gridMET specifies date as number of days since 1900-01-01. Here we use this
  # to assign a time field to our raster.
  terra::time(r) <- names(r) %>%
    stringr::str_split("=") %>%
    lapply(magrittr::extract2, 2) %>%
    unlist() %>%
    as.numeric() %>%
    lubridate::as_date(origin = as.Date("1900-01-01"))


  # If the variable is precipitation, take monthly sums, otherwise, use the mean.
  fun = ifelse(x == "pr", "sum", "mean")

  # Aggregate to a monthly timescale
  normals::aggregate_daily(
    r = r,
    variable = x,
    monthly = TRUE,
    agg_func = fun,
    filename = file.path(out_dir, glue::glue("{x}_monthly.tif"))
  )

  # Aggregate to an annual timescale
  normals::aggregate_daily(
    r = r,
    variable = x,
    monthly = FALSE,
    agg_func = fun,
    filename = file.path(out_dir, glue::glue("{x}_annual.tif"))
  )
})
```

Now that the monthly data are created, we can apply fit a gamma distribution to the monthly (or annual) data and create our climate normals! For more information on what is going on behind the hood here, read our information sidebar on the Montana Climate Normals Dashboard.

```r
summarized_dir <- file.path(data_dir, "processed", "montana", "summarized")
out_dir <- file.path(data_dir, "processed", "montana", "normals")

list.files(summarized_dir, full.names = T) %>%
  grep(".json", ., value = T, invert = T) %>%
  lapply(function(x) {
    meta <- stringr::str_split(basename(x), "_") %>% unlist()
    variable = meta[1]
    time = meta[2]
    r <- terra::rast(x)
    if (time == "monthly.tif") {
      lapply(1:12, function(month) {
        s <- terra::subset(r, which(lubridate::month(terra::time(r)) == month))
        gamma_from_rast(
          x = s,
          out_dir = out_dir,
          descriptor = glue::glue("{variable}_{tolower(month.abb[month])}"),
          rast_out = FALSE
        )
      })
    }  else {
      gamma_from_rast(
        x = r,
        out_dir = out_dir,
        descriptor = glue::glue("{variable}_annual"),
        rast_out = FALSE
      )
    }
  })
```

After running the above snippet of code, all the data will be saved out to the specified `out_dir` folder. For each variable and each timescale, there will be a file that gives the mean, median, variance, mode, alpha parameter and beta parameter. For example here is the mean annual precipitation for the Montana that was saved out to the folder (in inches):

```
r <- terra::rast(
  "~/data/mt_normals/processed/montana/normals/pr_annual_mean.tif"
)

# Convert from mm to inches
r <- r/25.4
terra::plot(r, mar = c(0,0,0,0))
```