Downloading Climate Data collected at SEUG Visitor Centers

Matthew W. Van Scoyoc

2020-10-26 15:11:38

Developed: 21 December 2017
Latest updated: 26 October 2020
Data files: downloaded from NOAA GHCND ftp site using rnoaa
Associated directories:
- "C:/R/Climate"
- "R:/NR_Climate/SEUG/Analysis (1B-Perm)/Climate Monitoring"
Notes: This script downloads data from NOAA's Global Historical Climatology Network (GHCN) then runs quality assurance-quality control procedures using the defined "flags" from NOAA. The data are then saved to an .RData files that is loaded in subsequent scripts.
Update notes: Included month data frame and formatting of wx.dat data frame that was happening at the beginning of subsequent scripts to increase efficiency of work flow.
Chunk headers were also added to make it possible to track errors when knitting to PDF using RMarkdown Notebooks.

```
# install.packages(c("rnoaa", "tidyverse", "lubridate", "knitr"))
library("rnoaa")
library("tidyverse")
```

```
## Registered S3 method overwritten by 'httr':
##   method           from
##   print.cache_info hoardr
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("lubridate")
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library("knitr")
```

Clear cache to ensure new data are downloaded. This step increases the time to run the script but ensures most recent data are downloaded.

```
meteo_clear_cache()
```

Save the date data were downloaded and create date stamp.

```
downloadDate <- today()
date.stamp <- paste(str_split(downloadDate, "-")[[1]], collapse = "")
```

Create a data frame of months.

```
month.df <- data.frame(month = 1:12,
                       waterMonth = c(4:12, 1:3),
                       mthName = factor(month.abb,
                                        levels = c(month.abb[10:12],
                                                   month.abb[1:9])),
                       season = c("Wi", "Wi", "Sp", "Sp", "Sp", "Su", "Su", "Su",
                                  "Fa", "Fa", "Fa", "Wi"))
```

Download station IDs and metadata from GHCN. This step takes some time to run, as the metadata file from GHCN is huge.

```
stations <- ghcnd_stations() %>% # Download station metadata from GHCN
  # Filter stations of interest
  filter(name %in% c("ARCHES NP HQS", "CANYONLANDS-THE NECK",
                     "CANYONLANDS-THE NEEDLES", "HANS FLAT RS", "HOVENWEEP NM",
                     "NATURAL BRIDGES NM", "MOAB")) %>%
  # Select variables of interest
  select(id, name, state, latitude, longitude, elevation) %>%
  distinct() %>% # Consolidate dataframe
  mutate(parkUnit = factor(c("ARCH", "CANY", "CANY", "CANY", "HOVE", "MOAB",
                             "NABR"),
                           levels = c("ARCH", "CANY", "HOVE", "NABR", "MOAB")),
         district = factor(c("ARCH", "ISKY", "NEED", "MAZE", "HOVE", "MOAB",
                             "NABR"),
                           levels = c("ARCH", "ISKY", "NEED", "MAZE", "HOVE",
                                      "NABR", "MOAB")),
         unitName = c("Arches National Park",
                      "Island in the Sky District, Canyonlands National Park",
                      "The Needles District, Canyonlands National Park",
                      "Hans Flat Ranger Station", "Hovenweep National Monument",
                      "Moab, Utah", "Natural Bridges National Monument"))
```

Download data from the GHCN ftp site.

```
dat.raw <- meteo_pull_monitors(monitors = stations$id, keep_flags = T,
                               var = c("PRCP", "TMAX", "TMIN"))
```

```
## file min/max dates: 1980-06-01 / 2020-10-31
## file min/max dates: 1965-06-01 / 2020-10-31
## file min/max dates: 1965-06-01 / 2020-10-31
## file min/max dates: 1980-10-01 / 2020-10-31
## file min/max dates: 1957-04-01 / 2020-10-31
## file min/max dates: 1893-01-01 / 2020-10-31
## file min/max dates: 1965-06-01 / 2020-10-31
```

Convert from wide to long data.

```
wx.dat <- dat.raw %>%
  select(id, date, prcp, tmax, tmin) %>%
  gather(element, value, 3:5, na.rm = F) %>%
  mutate(key = paste0(id, as.numeric(date), element)) %>%
  left_join(select(dat.raw, id, date, contains("mflag")) %>%
              gather(element, MFLAG, 3:5, na.rm = F) %>%
              separate(element, c("flag", "element"), "_") %>%
              mutate(key = paste0(id, as.numeric(date), element)) %>%
              select(key, MFLAG)) %>%
  left_join(select(dat.raw, id, date, contains("qflag")) %>%
              gather(element, QFLAG, 3:5, na.rm = F) %>%
              separate(element, c("flag", "element"), "_") %>%
              mutate(key = paste0(id, as.numeric(date), element)) %>%
              select(key, QFLAG)) %>%
  left_join(select(dat.raw, id, date, contains("sflag")) %>%
              gather(element, SFLAG, 3:5, na.rm = F) %>%
              separate(element, c("flag", "element"), "_") %>%
              mutate(key = paste0(id, as.numeric(date), element)) %>%
              select(key, SFLAG)) %>%
  left_join(select(stations, id, parkUnit, district)) %>%
  select(key, id, parkUnit, district, date, element, value, MFLAG, QFLAG, SFLAG)
```

```
## Joining, by = "key"
## Joining, by = "key"
## Joining, by = "key"
```

```
## Joining, by = "id"
```

Append dates within the *wx.data* data frame to the *stations* data frame.

```
stations <- stations %>%
  left_join(wx.dat %>%
              group_by(id) %>%
              summarise(startDate = min(date, na.rm = T),
                        endDate = max(date, na.rm = T),
                        timespan = as.duration(startDate %--% endDate)/dyears(1)))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## Joining, by = "id"
```

Load GHCND flag definitions from csv file.

```
flags <- read.csv("flags.csv", stringsAsFactors = F)
```

Read measurement flags (MFLAGs) in data.

```
mflags <- filter(flags, Type == "MFLAG") %>%
  rename(MFLAG = Flag)
wx.dat %>%
  filter(MFLAG %in% mflags$MFLAG) %>%
  left_join(mflags) %>%
  select(Definition) %>%
  distinct()
```

```
## Joining, by = "MFLAG"
```

```
## # A tibble: 2 x 1
```

```
##   Definition
##   <chr>
## 1 trace of precipitation, snowfall, or snow depth
## 2 identified as 'missing presumed zero' in DSI 3200 and 3206
```

Read quality flags (QFLAGs) in data.

```
qflags <- filter(flags, Type == "QFLAG") %>%
  rename(QFLAG = Flag)
wx.dat %>%
  filter(QFLAG %in% qflags$QFLAG) %>%
  left_join(qflags) %>%
  select(Definition) %>%
  distinct()
```

```
## Joining, by = "QFLAG"
```

```
## # A tibble: 7 x 1
##   Definition
##   <chr>
## 1 failed check on length of multiday period
## 2 failed climatological outlier check
## 3 failed internal consistency check
## 4 failed spatial consistency check
## 5 failed gap check
## 6 failed naught check
## 7 flagged as a result of an official Datzilla investigation
```

Read source flags (SFLAGs) in data.

```
sflags <- filter(flags, Type == "SFLAG") %>%
  rename(SFLAG = Flag)
wx.dat %>%
  filter(SFLAG %in% sflags$SFLAG) %>%
  left_join(sflags) %>%
  select(Definition) %>%
  distinct()
```

```
## Joining, by = "SFLAG"
```

```
## # A tibble: 3 x 1
##   Definition
##   <chr>
## 1 " U.S. Cooperative Summary of the Day (NCDC DSI-3200)"
## 2 " U.S. Cooperative Summary of the Day -- Transmitted via WxCoder3 (NCDC DSI-3~
## 3 " CDMP Cooperative Summary of the Day (NCDC DSI-3206)"
```

QAQC and format data.

```
wx.dat <- wx.dat %>%
  mutate(value = ifelse(QFLAG %in% qflags$QFLAG, NA, value),
         value = as.numeric(value),
         month = month(date),
         year = year(date),
         waterYr = ifelse(month >= 10, year + 1, year)) %>%
  left_join(month.df)
```

```
## Joining, by = "month"
```

Session information.

```
info <- sessionInfo()
r.ver <- paste(info$R.version$major, info$R.version$minor, sep=".")
```

Save data for subsequent scripts.

```
save(month.df, dat.raw, flags, stations, wx.dat, downloadDate, date.stamp, info,
     r.ver, file = "climateData.RData")
```

Session information.

```
print(info)
```

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 17763)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] knitr_1.30      lubridate_1.7.9 forcats_0.5.0   stringr_1.4.0
##  [5] dplyr_1.0.2     purrr_0.3.4     readr_1.4.0     tidyr_1.1.2
##  [9] tibble_3.0.4    ggplot2_3.3.2   tidyverse_1.3.0 rnoaa_1.2.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.5        utf8_1.1.4        assertthat_0.2.1  digest_0.6.25
##  [5] R6_2.4.1          cellranger_1.1.0  backports_1.1.10  reprex_0.3.0
##  [9] evaluate_0.14     httr_1.4.2        highr_0.8         pillar_1.4.6
## [13] rlang_0.4.8       curl_4.3          readxl_1.3.1      data.table_1.13.2
## [17] rstudioapi_0.11   blob_1.2.1        rmarkdown_2.5     urltools_1.7.3
## [21] triebeard_0.3.0   munsell_0.5.0     broom_0.7.2       compiler_4.0.3
## [25] modelr_0.1.8      xfun_0.18         pkgconfig_2.0.3   htmltools_0.5.0
## [29] tidyselect_1.1.0  gridExtra_2.3     httpcode_0.3.0    XML_3.99-0.5
## [33] fansi_0.4.1       crayon_1.3.4      hoardr_0.5.2      dbplyr_1.4.4
## [37] withr_2.3.0       rappdirs_0.3.1    crul_1.0.0        grid_4.0.3
## [41] jsonlite_1.7.1    gtable_0.3.0      lifecycle_0.2.0   DBI_1.1.0
## [45] magrittr_1.5      scales_1.1.1      cli_2.1.0         stringi_1.5.3
## [49] fs_1.5.0          xml2_1.3.2        ellipsis_0.3.1    generics_0.0.2
## [53] vctrs_0.3.4       tools_4.0.3       glue_1.4.2        hms_0.5.3
## [57] yaml_2.2.1        colorspace_1.4-1  rvest_0.3.6       haven_2.3.1
```