

Mesonet Downloads

* Download data from Oklahoma Mesonet

- Multiple years can be downloaded by modifying “annis”
- Smaller periods can be downloaded using alternate script
- Totals for day are stored in next day @ 00:00 so get an extra day
- Uses year range and current time as timestamp in file name to prevent overwriting previous data.
- Writes a dat file for each year - this is a normal csv but excel will choke on opening it, which prevents excel from jacking up the date/time fields. Can still be imported into excel as a text file - or just change the extension to csv
- When attempting to download several years of data there is a high likelihood of receiving an HTTP 500 error - this error comes from the server - there is no fixing it. The script will throw an error while downloading a single day's data but will recover and continue. Make a note of which years had issues, change “annis” to reflect those years and rerun script. Depending on how many years are involved this may take several iterations. There may be files written to disk with partial data that represent those years. Error message looks like this: *cannot open URL 'http://www.mesonet.org/index.php/dataMdfMts/dataController/getFile/20110526burn/mts/TEXT/': HTTP status was '500 Internal Server Error'Error in file(file, "rt") : cannot open the connection* The data from 20110526 didn't get retrieved. Rerunning the script for 2011 is the solution - even then it might not work the first go around.
- Check timezone to make sure it is right - high temps should occur around 1400 hrs or thereabouts

```
library(tidyverse)
library(lubridate)
library(magrittr)
library(rprojroot)
library("tidylog", warn.conflicts = FALSE)
library(tictoc)
library(okmesonet)## Interface for Oklahoma Mesonet http://www.mesonet.org/
                  ## https://cran.r-project.org/package=okmesonet

# 2019-12-03 09:27:41 -----mdp
# 2019-12-04 08:14:54 -----mdp
# 2019-12-10 08:19:35 -----mdp
# 2020-06-23 12:34:34 -----mdp
# 2020-06-24 12:06:26 -----mdp
# 2020-07-13 14:38:28 -----mdp
# 2020-09-03 10:58:15 -----mdp # fields weren't the same in both methods
# 2020-09-24 12:12:07 -----mdp # now() in file name
# 2021-01-07 15:22:48 -----mdp converted to rmarkdown

## Local stuff =====
base_path  <- find_rstudio_root_file() ## Finds path for project
dat_path   <- file.path(base_path, "dat_output/")
```

List all Oklahoma Mesonet stations and locations.

Only for looking up station id.

```
okstations
```

Display current conditions for all sites. FYI only - not required *Arrows to scroll columns are in column heading.*

```
curobs <- okcurobs(localtime = TRUE)
curobs
```

Enter station Identifier - all caps and with the quotes

```
station_code <- "BURN"
```

Download multiple years

Modify and run one of the lines below If time span involves Dec 31, the precipitation value for that date will be in Jan 1 of next year - Just get both years - code will adjust for current date. You may have to delete some extra data on the end. For current year, code subtracts 2 days from current date and uses that for end date - avoids trying to get data that doesn't exist. As discussed above this is also where you would change values for years that don't download successfully.

```
#annis <- c(2011, 2012, 2019, 2020)
#annis <- c(2011) ## years that don't complete on first go around
#annis <- c(2019:2020)
annis <- c(2020:2021)
```

Theoretically, multiple stations could be run through a loop but wouldn't be likely to finish successfully anyway so I didn't attempt it. Change stations manually.

Variables available and definitions are here: http://www.mesonet.org/index.php/site/about/mdf_mts_files Temperature, precipitation, and wind values are @ 5 minute intervals. Other values like soil temps differ in frequency. There are additional variables available that are not included below.

Change desired variables below Limiting number of variables doesn't have much effect on speed. I think the okkmesonet package still has to download the entire mts file for a particular day and discards other variables.

```
# meso_variables <- c("RAIN", "TAIR", "RELH", "WSPD", "WVEC",
#                    "WDIR", "SRAD", "WS2M", "TS05", "TS10",
#                    "TB10", "TR05", "WMAX", "PRES", "WDIR")

#meso_variables <- c("RAIN", "TAIR", "TS05", "TS10")
meso_variables <- c("TAIR", "TS10", "TB10")
```

This block does the heavy lifting

The output represents the beginning and ending date for each year and the file name that data was written to, along with the time elapsed for processing each year and for all years. Watch for missing years and rerun for those years.

```
tic() ## This takes about 2 min/year with 5 sec wait between - single periods can be run below
for (i in annis) {
```

```
  b_bin <- ymd_hms(paste0(i, "-01-01 00:00:00")) ## daily total ppt is stored in
```

```

b_end <- ymd_hms(paste0(i, "-12-31 00:00:00")) ## next day @ 00:00 so you may need
## one value from next month

if (format(b_end, "%Y") == year(Sys.Date())) { ## If year == current year, subtract 2 days
  ## from current date so we don't try to
  ## get values that don't exist yet
  b_end <- now() - ddays(2) ## Could not do this with if_else - wouldn't
  ## evaluate conditional
} ##
print(b_gin)
tic()

try({## Try lets script continue after error - you can go back and rerun what timed out
  ## the first time by changing the values in annis above

  df <- okmts(beginntime = b_gin, ## Retrieve specific fields
    endtime = b_end,
    station = station_code, ## see okstations
    missingNA = TRUE,
    localtime = TRUE,
    variables = meso_variables)

  write_csv(df, paste0(dat_path, station_code, "_Mesonet_",
    year(b_gin), "_", format(now(), "%Y%m%d_%H%M"), ".dat"))

})

print(b_end)
print(head(df, 10))
print(paste0(dat_path, station_code, "_Mesonet_",
  year(b_gin), "_", format(now(), "%Y%m%d_%H%M"), ".dat"))

toc()
Sys.sleep(5)
}
toc()

```

Download a single period

Change begining (b_gin) and ending (b_end) dates for desired period. Attempting to download periods longer than one year are likely to fail due to http 500 error mentioned above.

```

b_gin <- ymd_hms("2020-04-29 12:00:00")
b_end <- ymd_hms("2020-05-01 12:00:00")

```

Chose station identifier - see table above. Uncomment one of existing or modify for different location.

```

station_code <- "BURN"
#station_code <- "ARD2"

```

Edit variables - see url above for documentation If replacing missed periods caused by above server error there's no need to run this block unless file has been closed or session restarted. This is the same as code at line 105

```
# meso_variables <- c("RAIN", "TAIR", "RELH", "WSPD", "WVEC",
#                    "WDIR", "SRAD", "WS2M", "TS05", "TS10",
#                    "TB10", "TRO5", "WMAX", "PRES", "WDIR")
```

```
meso_variables <- c("RAIN", "TAIR") # This can be assigned in code block in previous section as well
```

Download data and write file. Output at bottom is start and end date as well as name and path of generated file.

```
tic()
df <- okmts(begintime = b_gin,
            endtime   = b_end,
            station   = station_code, ## Pick station ID
            missingNA = TRUE,
            localtime = TRUE,
            variables = meso_variables)
write_csv(df, paste0(dat_path, station_code, "_Mesonet_", year(b_gin), "_",
                                                             month(b_gin), "_",
                                                             format(now(), "%Y%m%d_%H%M"), ".dat"))

toc()

print(paste0("Start date = ", b_gin))
print(paste0("End date   = ", b_end))
print(paste0(dat_path, station_code, "_Mesonet_", year(b_gin), "_",
                                                             month(b_gin), "_",
                                                             format(now(), "%Y%m%d_%H%M"), ".dat"))
```

Mesonet_Processing.R can be used to compile resulting files into one and do some unit / formula conversion depending on what fields were downloaded. *Soil_Temps.Rmd* compiles data and creates some graphs of soil temperatures under vegetation and bare ground using data derived from these files.

```
# Generates package citations
library(grateful)
cites <- get_citations(scan_packages())
```

```
cites
```

```
# There is an entry in the YAML as well that makes this work
```

References

- Allred, Brady, Torre Hovick, and Samuel Fuhlendorf. 2014. *Okmesonet: Retrieve Oklahoma Mesonet Climatological Data*. <https://CRAN.R-project.org/package=okmesonet>.
- Bache, Stefan Milton, and Hadley Wickham. 2020. *Magrittr: A Forward-Pipe Operator for r*. <https://CRAN.R-project.org/package=magrittr>.
- Bååth, Rasmus. 2018. *Beepr: Easily Play Notification Sounds on Any Platform*. <https://CRAN.R-project.org/package=beepr>.
- Elbers, Benjamin. 2020. *Tidylog: Logging for 'Dplyr' and 'Tidyr' Functions*. <https://CRAN.R-project.org/package=tidylog>.

- Grolemund, Garrett, and Hadley Wickham. 2011. “Dates and Times Made Easy with lubridate.” *Journal of Statistical Software* 40 (3): 1–25. <https://www.jstatsoft.org/v40/i03/>.
- Izrailev, Sergei. 2021. *Tictoc: Functions for Timing r Scripts, as Well as Implementations of Stack and List Structures*. <https://CRAN.R-project.org/package=tictoc>.
- Müller, Kirill. 2020. *Rprojroot: Finding Files in Project Subdirectories*. <https://CRAN.R-project.org/package=rprojroot>.
- R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.
- Rodríguez-Sánchez, Francisco, and Shaurita D. Hutchins. 2020. *Grateful: Facilitate Citation of r Packages*. <https://github.com/Pakillo/grateful>.
- Simonsohn, Uri, and Hugo Gruson. 2021. *Groundhog: The Simplest Solution to Version-Control for CRAN Packages*. <https://CRAN.R-project.org/package=groundhog>.
- Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D’Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. “Welcome to the tidyverse.” *Journal of Open Source Software* 4 (43): 1686. <https://doi.org/10.21105/joss.01686>.
- Wickham, Hadley, and Jim Hester. 2021. *Readr: Read Rectangular Text Data*. <https://CRAN.R-project.org/package=readr>.
- Xie, Yihui, Christophe Dervieux, and Emily Riederer. 2020. *R Markdown Cookbook*. Boca Raton, Florida: Chapman; Hall/CRC. <https://bookdown.org/yihui/rmarkdown-cookbook>.