```
library(rgdal)
```

```
## Loading required package:  sp
## rgdal:  version:  1.5-19, (SVN revision 1092)
## Geospatial Data Abstraction Library extensions to R successfully
loaded
## Loaded GDAL runtime:  GDAL 1.11.4, released 2016/01/25
## Path to GDAL shared files:  /usr/share/gdal
## GDAL binary built with GEOS: TRUE
## Loaded PROJ runtime:  Rel.  4.8.0, 6 March 2012, [PJ_VERSION: 480]
## Path to PROJ shared files:  (autodetected)
## Linking to sp version:1.3-1
```

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------- tidyverse
1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts()
--
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(plyr)
```

```
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr
first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
## Attaching package:  'plyr'
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
## The following object is masked from 'package:purrr':
##
##     compact
```

```
library(rnoaa)
```

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



```
library(ncdf4)
```

Obtaining Climate Records

Marc Los Huertos

March 18, 2022

1 Terrestrial Meteorological Data

1.1 Selected History of Meteorology

1.1.1 List of Cities

rNOAA has a simple function to list the cities:
Commented out – takes forever and errors out!

```
ncdc_locs(locationcategoryid='CITY', sortfield='name', sortorder='desc')
```

```
## $meta
## $meta$totalCount
## [1] 1989
##
## $meta$pageCount
## [1] 25
##
## $meta$offset
## [1] 1
##
##
## $data
##       mindate    maxdate                 name datacoverage           id
## 1  1892-08-01 2021-12-20           Zwolle, NL       1.0000 CITY:NL000012
## 2  1901-01-01 2022-03-13           Zurich, SZ       1.0000 CITY:SZ000007
## 3  1957-07-01 2022-03-13         Zonguldak, TU       1.0000 CITY:TU000057
## 4  1906-01-01 2022-03-13            Zinder, NG       0.9025 CITY:NG000004
## 5  1973-01-01 2022-03-13        Ziguinchor, SG       1.0000 CITY:SG000004
## 6  1938-01-01 2022-03-13         Zhytomyra, UP       0.9723 CITY:UP000025
## 7  1948-03-01 2022-03-13        Zhezkazgan, KZ       0.9302 CITY:KZ000017
## 8  1951-01-01 2022-03-13         Zhengzhou, CH       1.0000 CITY:CH000045
## 9  1941-01-01 2022-03-13          Zaragoza, SP       1.0000 CITY:SP000021
## 10 1936-01-01 2009-06-17      Zaporiyhzhya, UP       1.0000 CITY:UP000024
## 11 1957-01-01 2022-03-13          Zanzibar, TZ       0.8016 CITY:TZ000019
```

```
## 12 1973-01-01 2022-03-13            Zanjan, IR       0.9105 CITY:IR000020
## 13 1893-01-01 2022-03-15     Zanesville, OH US       1.0000 CITY:US390029
## 14 1912-01-01 2022-03-13             Zahle, LE       0.9819 CITY:LE000004
## 15 1951-01-01 2022-03-13           Zahedan, IR       0.9975 CITY:IR000019
## 16 1860-12-01 2022-03-13            Zagreb, HR       1.0000 CITY:HR000002
## 17 1929-07-01 2022-02-01         Zacatecas, MX       1.0000 CITY:MX000036
## 18 1947-01-01 2022-03-13 Yuzhno-Sakhalinsk, RS       1.0000 CITY:RS000081
## 19 1893-01-01 2022-03-15           Yuma, AZ US       1.0000 CITY:US040015
## 20 1942-02-01 2022-03-14   Yucca Valley, CA US       1.0000 CITY:US060048
## 21 1885-01-01 2022-03-15      Yuba City, CA US       1.0000 CITY:US060047
## 22 1998-02-01 2022-03-13            Yozgat, TU       0.9993 CITY:TU000056
## 23 1893-01-01 2022-03-15     Youngstown, OH US       1.0000 CITY:US390028
## 24 1894-01-01 2022-03-15           York, PA US       1.0000 CITY:US420024
## 25 1869-01-01 2022-03-15        Yonkers, NY US       1.0000 CITY:US360031
##
## attr(,"class")
## [1] "ncdc_locs"

ncdc_locs(locationcategoryid='ST', limit=52) # States

## $meta
## $meta$totalCount
## [1] 51
##
## $meta$pageCount
## [1] 52
##
## $meta$offset
## [1] 1
##
##
## $data
##       mindate    maxdate                name datacoverage      id
## 1  1888-02-01 2022-03-15             Alabama            1 FIPS:01
## 2  1893-09-01 2022-03-15              Alaska            1 FIPS:02
## 3  1867-08-01 2022-03-15             Arizona            1 FIPS:04
## 4  1871-07-01 2022-03-15            Arkansas            1 FIPS:05
## 5  1850-10-01 2022-03-15          California            1 FIPS:06
## 6  1852-10-01 2022-03-15            Colorado            1 FIPS:08
## 7  1884-11-01 2022-03-15         Connecticut            1 FIPS:09
## 8  1893-01-01 2022-03-15            Delaware            1 FIPS:10
## 9  1870-11-01 2022-03-14 District of Columbia            1 FIPS:11
## 10 1871-09-12 2022-03-15             Florida            1 FIPS:12
## 11 1849-01-01 2022-03-15             Georgia            1 FIPS:13
## 12 1905-01-01 2022-03-15              Hawaii            1 FIPS:15
## 13 1892-06-01 2022-03-15               Idaho            1 FIPS:16
```

```
## 14 1870-10-15 2022-03-15            Illinois           1 FIPS:17
## 15 1886-02-01 2022-03-15             Indiana           1 FIPS:18
## 16 1888-06-01 2022-03-15                Iowa           1 FIPS:19
## 17 1857-04-01 2022-03-15              Kansas           1 FIPS:20
## 18 1872-10-01 2022-03-15            Kentucky           1 FIPS:21
## 19 1882-07-01 2022-03-15           Louisiana           1 FIPS:22
## 20 1885-06-01 2022-03-15               Maine           1 FIPS:23
## 21 1880-01-01 2022-03-15            Maryland           1 FIPS:24
## 22 1831-02-01 2022-03-15       Massachusetts           1 FIPS:25
## 23 1887-06-01 2022-03-15            Michigan           1 FIPS:26
## 24 1886-01-01 2022-03-15           Minnesota           1 FIPS:27
## 25 1891-01-01 2022-03-15         Mississippi           1 FIPS:28
## 26 1890-01-01 2022-03-15            Missouri           1 FIPS:29
## 27 1891-08-01 2022-03-15             Montana           1 FIPS:30
## 28 1878-01-01 2022-03-15            Nebraska           1 FIPS:31
## 29 1877-07-01 2022-03-15              Nevada           1 FIPS:32
## 30 1868-01-01 2022-03-15       New Hampshire           1 FIPS:33
## 31 1865-06-01 2022-03-15          New Jersey           1 FIPS:34
## 32 1870-01-01 2022-03-15          New Mexico           1 FIPS:35
## 33 1869-01-01 2022-03-15            New York           1 FIPS:36
## 34 1869-03-01 2022-03-15      North Carolina           1 FIPS:37
## 35 1891-07-01 2022-03-15        North Dakota           1 FIPS:38
## 36 1871-01-01 2022-03-15                Ohio           1 FIPS:39
## 37 1870-04-01 2022-03-15            Oklahoma           1 FIPS:40
## 38 1871-11-01 2022-03-15              Oregon           1 FIPS:41
## 39 1849-04-01 2022-03-15        Pennsylvania           1 FIPS:42
## 40 1893-01-01 2022-03-15        Rhode Island           1 FIPS:44
## 41 1849-05-01 2022-03-15      South Carolina           1 FIPS:45
## 42 1893-01-01 2022-03-15        South Dakota           1 FIPS:46
## 43 1879-01-01 2022-03-15           Tennessee           1 FIPS:47
## 44 1852-04-01 2022-03-15               Texas           1 FIPS:48
## 45 1887-12-01 2022-03-15                Utah           1 FIPS:49
## 46 1883-12-01 2022-03-15             Vermont           1 FIPS:50
## 47 1869-01-01 2022-03-15            Virginia           1 FIPS:51
## 48 1856-01-01 2022-03-15          Washington           1 FIPS:53
## 49 1854-01-01 2022-03-15       West Virginia           1 FIPS:54
## 50 1869-01-01 2022-03-15           Wisconsin           1 FIPS:55
## 51 1889-01-01 2022-03-15             Wyoming           1 FIPS:56
##
## attr(,"class")
## [1] "ncdc_locs"

ncdc_locs(locationid='FIPS:01', limit=52) # Alabama

## $meta
## NULL
```

```
##
## $data
##      mindate    maxdate    name datacoverage      id
## 1 1888-02-01 2022-03-15 Alabama            1 FIPS:01
##
## attr(,"class")
## [1] "ncdc_locs"
```

```
ncdc_locs(locationcategoryid='CITY', locationid='FIPS:01', sortfield='name', sortorder='desc
```

```
## $meta
## NULL
##
## $data
##      mindate    maxdate    name datacoverage      id
## 1 1888-02-01 2022-03-15 Alabama            1 FIPS:01
##
## attr(,"class")
## [1] "ncdc_locs"
```

```
ncdc_datasets(locationcategoryid='CITY', locationid='FIPS:01', sortfield='name', sortorder='
```

```
## $meta
## $meta$offset
## [1] 1
##
## $meta$count
## [1] 11
##
## $meta$limit
## [1] 25
##
##
## $data
##                      uid    mindate    maxdate                         name
## 1  gov.noaa.ncdc:C00708 1994-05-20 2022-03-13   Weather Radar (Level III)
## 2  gov.noaa.ncdc:C00345 1991-06-05 2022-03-14    Weather Radar (Level II)
## 3  gov.noaa.ncdc:C00313 1900-01-01 2014-01-01        Precipitation Hourly
## 4  gov.noaa.ncdc:C00505 1970-05-12 2014-01-01     Precipitation 15 Minute
## 5  gov.noaa.ncdc:C00822 2010-01-01 2010-12-01             Normals Monthly
## 6  gov.noaa.ncdc:C00824 2010-01-01 2010-12-31              Normals Hourly
## 7  gov.noaa.ncdc:C00823 2010-01-01 2010-12-31               Normals Daily
## 8  gov.noaa.ncdc:C00821 2010-01-01 2010-01-01     Normals Annual/Seasonal
## 9  gov.noaa.ncdc:C00947 1763-01-01 2022-01-01  Global Summary of the Year
## 10 gov.noaa.ncdc:C00946 1763-01-01 2022-03-01 Global Summary of the Month
## 11 gov.noaa.ncdc:C00861 1763-01-01 2022-03-14             Daily Summaries
```

```
##    datacoverage         id
## 1          0.95    NEXRAD3
## 2          0.95    NEXRAD2
## 3          1.00 PRECIP_HLY
## 4          0.25  PRECIP_15
## 5          1.00 NORMAL_MLY
## 6          1.00 NORMAL_HLY
## 7          1.00 NORMAL_DLY
## 8          1.00 NORMAL_ANN
## 9          1.00       GSOY
## 10         1.00       GSOM
## 11         1.00      GHCND
##
## attr(,"class")
## [1] "ncdc_datasets"
```

The function queries the NOAA website and retrieves city names and the dates of the climate records. Importantly, the records include the station ID, which can be used to college the data for that city.

NOTE2: It would be nice to make a map of how concentrated the stations spatially.

1.2 Getting Data

1.2.1 Selection Stations

```
ncdc_stations(datasetid='GHCND', locationid='FIPS:12017', stationid='GHCND:USC00084289')
```

```
## $meta
## NULL
##
## $data
##   elevation    mindate    maxdate latitude                  name datacoverage
## 1      17.7 1899-01-01 2022-03-04 28.80286 INVERNESS 3 SE, FL US            1
##                   id elevationUnit longitude
## 1 GHCND:USC00084289        METERS -82.31266
##
## attr(,"class")
## [1] "ncdc_stations"
```

```
# alabama stations.. sorted by the most recent
test <- ncdc_stations(datasetid='GHCND', locationid='FIPS:01', limit=1000, sortfield = 'maxd
test <- ncdc_stations(datasetid='GSOM', locationid='FIPS:01', limit=1000, sortfield = 'maxda

str(test)
```

```
## List of 2
##  $ meta:List of 3
##   ..$ totalCount: int 996
##   ..$ pageCount : int 1000
##   ..$ offset    : int 1
##  $ data:'data.frame': 996 obs. of  9 variables:
##   ..$ elevation    : num [1:996] 86.8 216.4 324 214 182 ...
##   ..$ mindate      : chr [1:996] "2006-09-01" "2003-10-01" "2008-06-01" "2015-01-01" ...
##   ..$ maxdate      : chr [1:996] "2022-03-01" "2022-02-01" "2022-02-01" "2022-02-01" ...
##   ..$ latitude     : num [1:996] 32 33.1 34.3 34.7 32.6 ...
##   ..$ name         : chr [1:996] "EUFAULA WEEDON FIELD AIRPORT, AL US" "WADLEY NR 2, AL U
##   ..$ datacoverage : num [1:996] 0.882 0.928 0.988 0.838 0.922 ...
##   ..$ id           : chr [1:996] "GHCND:USW00063872" "GHCND:USC00018608" "GHCND:US1ALMS0C
##   ..$ elevationUnit: chr [1:996] "METERS" "METERS" "METERS" "METERS" ...
##   ..$ longitude    : num [1:996] -85.1 -85.6 -86.2 -86.8 -85.5 ...
##  - attr(*, "class")= chr "ncdc_stations"

recent = test$data[test$data$maxdate=='2022-02-01',]
str(recent)

## 'data.frame': 251 obs. of  9 variables:
##  $ elevation    : num  216 324 214 182 245 ...
##  $ mindate      : chr  "2003-10-01" "2008-06-01" "2015-01-01" "2019-01-01" ...
##  $ maxdate      : chr  "2022-02-01" "2022-02-01" "2022-02-01" "2022-02-01" ...
##  $ latitude     : num  33.1 34.3 34.7 32.6 34.1 ...
##  $ name         : chr  "WADLEY NR 2, AL US" "ALBERTVILLE 5.5 N, AL US" "MADISON 3.2 W, AI
##  $ datacoverage : num  0.928 0.988 0.838 0.922 0.898 ...
##  $ id           : chr  "GHCND:USC00018608" "GHCND:US1ALMS0022" "GHCND:US1ALLS0026" "GHCNI
##  $ elevationUnit: chr  "METERS" "METERS" "METERS" "METERS" ...
##  $ longitude    : num  -85.6 -86.2 -86.8 -85.5 -87.4 ...

(longest = recent[recent$mindate == min(recent$mindate),])

##    elevation    mindate    maxdate latitude              name datacoverage
## 15      75.9 1892-03-01 2022-02-01 32.69212 GREENSBORO, AL US       0.9539
##                   id elevationUnit longitude
## 15 GHCND:USC00013511        METERS -87.57603

(startyear=as.numeric(format(as.Date(longest$mindate), format = "%Y")))

## [1] 1892

#(endyear=as.numeric(format(as.Date(longest£maxdate), format = "%Y")))
ncdc(datasetid='GSOM', stationid=longest$id, startdate = '2021-01-01', enddate = '2022-01-01

## $meta
## $meta$totalCount
```

```
## [1] 13
##
## $meta$pageCount
## [1] 25
##
## $meta$offset
## [1] 1
##
##
## $data
## # A tibble: 13 x 8
##    date                datatype station           value fl_a  fl_M  fl_Q  fl_S
##    <chr>               <chr>    <chr>             <dbl> <chr> <chr> <chr> <chr>
##  1 2021-01-01T00:00:00 TMAX     GHCND:USC00013511  13.3 ""    ""    ""    7
##  2 2021-02-01T00:00:00 TMAX     GHCND:USC00013511  13.9 ""    ""    ""    7
##  3 2021-03-01T00:00:00 TMAX     GHCND:USC00013511  21.3 ""    ""    ""    7
##  4 2021-04-01T00:00:00 TMAX     GHCND:USC00013511  23.2 ""    ""    ""    7
##  5 2021-05-01T00:00:00 TMAX     GHCND:USC00013511  26.7 ""    ""    ""    7
##  6 2021-06-01T00:00:00 TMAX     GHCND:USC00013511  29.9 ""    ""    ""    7
##  7 2021-07-01T00:00:00 TMAX     GHCND:USC00013511  30.8 ""    ""    ""    7
##  8 2021-08-01T00:00:00 TMAX     GHCND:USC00013511  31.6 "1"   ""    ""    7
##  9 2021-09-01T00:00:00 TMAX     GHCND:USC00013511  28.0 ""    ""    ""    7
## 10 2021-10-01T00:00:00 TMAX     GHCND:USC00013511  24.6 ""    ""    ""    7
## 11 2021-11-01T00:00:00 TMAX     GHCND:USC00013511  17.7 ""    ""    ""    7
## 12 2021-12-01T00:00:00 TMAX     GHCND:USC00013511  19.6 ""    ""    ""    7
## 13 2022-01-01T00:00:00 TMAX     GHCND:USC00013511  12.8 ""    ""    ""    7
##
## attr(,"class")
## [1] "ncdc_data"
```

1.2.2 Functions to Collect and Clean GSOM

```
get_GSOM <- function(stid, datatype) {
   wtr<-list()  # create an empty list
   for (i in startyear:2021) {
      start_date <- paste0(i, "-01-01")
      end_date <- paste0(i, "-12-31")

      #save data portion to the list (elements named for the year
      wtr[[as.character(i)]] <- ncdc(datasetid='GSOM', stationid=stid, datatypeid=datatype,
   }
   #return the full list of data frames
   return(wtr)
}
```

```r
GSOM_TMAX <- get_GSOM(longest$id, 'TMAX')
GSOM_TMIN <- get_GSOM(longest$id, 'TMIN')

#bind the dataframes in the list together into one large dataframe
library(dplyr)
tbl_TMAX <- dplyr::bind_rows(GSOM_TMAX)
tbl_TMIN <- dplyr::bind_rows(GSOM_TMIN)

class(test) # [1] "tbl_df"     "tbl"        "data.frame"

## [1] "ncdc_stations"

dfTbl_TMAX = as.data.frame(tbl_TMAX)
dfTbl_TMIN = as.data.frame(tbl_TMIN)
class(dfTbl_TMAX) # [1] "data.frame"

## [1] "data.frame"

dfTbl_TMAX$TMAX = dfTbl_TMAX$value/10*9/5+32
dfTbl_TMIN$TMIN = dfTbl_TMIN$value/10*9/5+32

dfTbl_TMAX$Date = as.Date(dfTbl_TMAX$date)
dfTbl_TMIN$Date = as.Date(dfTbl_TMIN$date)

dfTbl_TMAX <- subset(dfTbl_TMAX, select=c(Date, station, TMAX))
dfTbl_TMIN <- subset(dfTbl_TMIN, select=c(Date, station, TMIN))
str(dfTbl_TMIN)

## 'data.frame': 1486 obs. of  3 variables:
##  $ Date   : Date, format: "1892-03-01" "1892-04-01" ...
##  $ station: chr  "GHCND:USC00013511" "GHCND:USC00013511" "GHCND:USC00013511" "GHCND:USC0C
##  $ TMIN   : num  32.9 34.3 34.9 35.6 35.9 ...

GSOM <- merge(dfTbl_TMAX, dfTbl_TMIN, by="Date")

GSOM$Month = as.numeric(format(as.Date(GSOM$Date), format = "%m"))
GSOM$Year = as.numeric(format(as.Date(GSOM$Date), format = "%Y"))
```

```r
# find most important month
sumstats = NA
for (m in 1:12){
  TMAX.lm = lm(TMAX~Date, GSOM[GSOM$Month==m,])
  TMIN.lm = lm(TMIN~Date, GSOM[GSOM$Month==m,])
sumstats = rbind(sumstats, data.frame(Month = m, TMIN_Slope = coef(TMIN.lm)[2], TMIN_r2 = su
```

```
}
sumstats=data.frame(sumstats)[-1,]
(maxmonth = sumstats$Month[sumstats$TMIN_Slope == max(sumstats$TMIN_Slope, na.rm=T)])
(maxmonth = sumstats$Month[sumstats$TMAX_Slope == max(sumstats$TMAX_Slope, na.rm=T)])

par(las=1, mfrow=c(2,1), mar= c(2, 4, 2, 1) + 0.1)

for(i in min(GSOM$Year+5):max(GSOM$Year), by=3)
   {
   GSOMsub <- GSOM[GSOM$Year<=i,]
plot(TMIN~Date, GSOMsub[GSOMsub$Month==maxmonth,], col='gray50', pch=20, xlab="")
GSOM.lm = lm(TMIN~Date, GSOMsub[GSOMsub$Month==maxmonth,])
abline(coef(GSOM.lm), col='red')
summary(GSOM.lm); anova(GSOM.lm)$'Pr(>F)'[1]
plot(TMAX~Date, GSOMsub[GSOMsub$Month==maxmonth,], col='gray50', pch=20, xlab="")
GSOM.lm = lm(TMAX~Date, GSOMsub[GSOMsub$Month==maxmonth,])
abline(coef(GSOM.lm), col='red')
#summary(GSOM.lm);
text(i, coef(GSOM.lm)[2]*i+coef(GSOM.lm)[1], paste("p-value", round(anova(GSOM.lm)$'Pr(>F)'[
}

## Error:  <text>:16:41:  unexpected ','
## 15:
## 16:  for(i in min(GSOM$Year+5):max(GSOM$Year),
##                                              ^
```

```
library(magick)

## Linking to ImageMagick 6.9.10.68
## Enabled features:  cairo, fontconfig, freetype, ghostscript, lcms,
pango, rsvg, x11
## Disabled features:  fftw, heic, raw, webp
## Using 48 threads

#setwd("/home/CAMPUS/mwl04747/github/Climate_Change_Narratives/docs/Social_Media")

par(las=1, mfrow=c(2,1), mar= c(2, 4, 2, 1) + 0.1)
img <- image_graph(600, 480, res = 96)
for(i in min(GSOM$Year+5):max(GSOM$Year)) {
#png(paste0("png//Rplot_", i, ".png"), width = 480, height = 480, units = "px", pointsize =
   GSOMsub <- GSOM[GSOM$Year<=i,]
par(las=1, mfrow=c(2,1), mar= c(2, 4, 2, 1) + 0.1)
plot(TMIN~Date, GSOMsub[GSOMsub$Month==maxmonth,], col='gray50', pch=20, xlab="")
GSOM.lm = lm(TMIN~Date, GSOMsub[GSOMsub$Month==maxmonth,])
abline(coef(GSOM.lm), col='red')
```

```
#summary(GSOM.lm); anova(GSOM.lm)£'Pr(>F)'[1]
plot(TMAX~Date, GSOMsub[GSOMsub$Month==maxmonth,], col='gray50', pch=20, xlab="")
GSOM.lm = lm(TMAX~Date, GSOMsub[GSOMsub$Month==maxmonth,])
abline(coef(GSOM.lm), col='red')
#summary(GSOM.lm);
text(i, coef(GSOM.lm)[2]*i+coef(GSOM.lm)[1], paste("p-value", round(anova(GSOM.lm)$'Pr(>F)'[
}
```

```
## Error in `[.data.frame`(GSOMsub, GSOMsub$Month == maxmonth, ):  object
'maxmonth' not found
```

```
dev.off()
```

```
## pdf
##   2
```

```
GSOM_animation <- image_animate(img, fps = 1, optimize = TRUE)
print(GSOM_animation)
```

```
## # A tibble: 0 x 7
## # ... with 7 variables: format <chr>, width <int>, height <int>,
## #   colorspace <chr>, matte <lgl>, filesize <int>, density <chr>
```

Other attempts...

```
out <- ncdc(datasetid='NORMAL_DLY', stationid='GHCND:USW00014895', datatypeid='dly-tmax-norm
```

```
with_units <- ncdc(datasetid='GHCND', stationid='GHCND:USW00014895', datatypeid='PRCP', star
head( with_units$data )
```

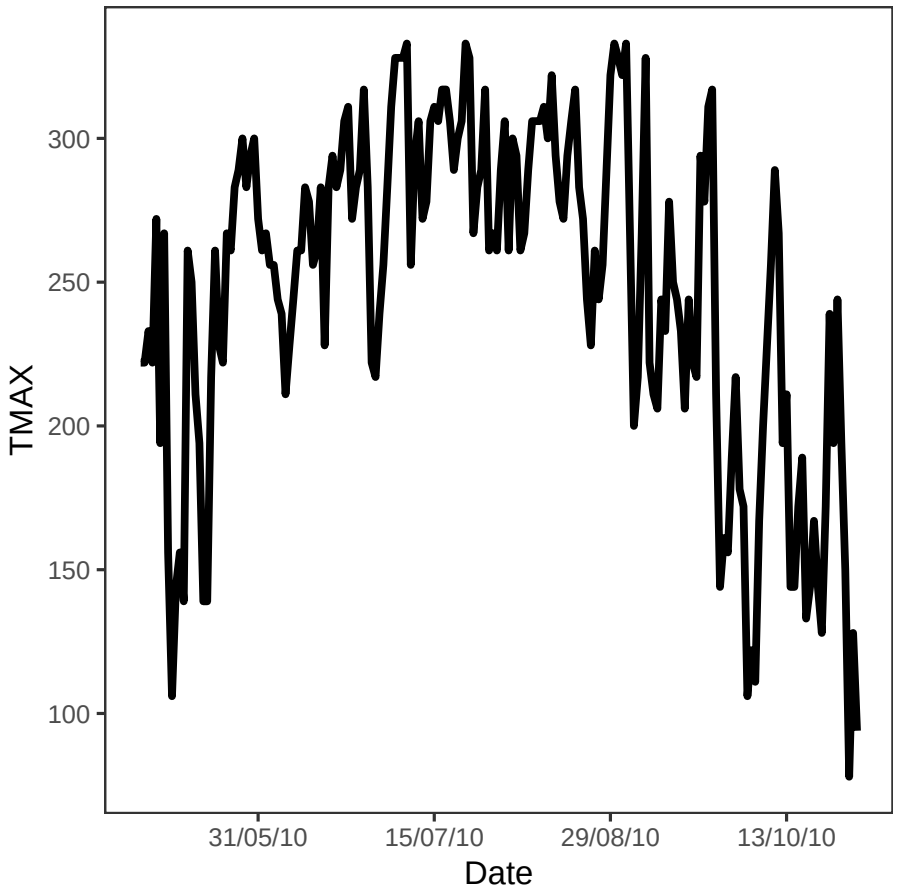
```
## # A tibble: 6 x 9
##   date                datatype station      value fl_m  fl_q  fl_so fl_t  units
##   <chr>               <chr>    <chr>        <int> <chr> <chr> <chr> <chr> <chr>
## 1 2010-05-01T00:00:00 PRCP     GHCND:USW00~     0 "T"   ""    0     2400  mm_te~
## 2 2010-05-02T00:00:00 PRCP     GHCND:USW00~    30 ""    ""    0     2400  mm_te~
## 3 2010-05-03T00:00:00 PRCP     GHCND:USW00~    51 ""    ""    0     2400  mm_te~
## 4 2010-05-04T00:00:00 PRCP     GHCND:USW00~     0 "T"   ""    0     2400  mm_te~
## 5 2010-05-05T00:00:00 PRCP     GHCND:USW00~    18 ""    ""    0     2400  mm_te~
## 6 2010-05-06T00:00:00 PRCP     GHCND:USW00~    30 ""    ""    0     2400  mm_te~
```

```
with_units <- ncdc(datasetid='GHCND', stationid='GHCND:USW00014895', datatypeid='TMAX', star
head( with_units$data )
```

```
## # A tibble: 6 x 9
##   date                datatype station      value fl_m  fl_q  fl_so fl_t  units
```

```
##   <chr>               <chr>    <chr>       <int> <chr> <chr> <chr> <chr> <chr>
## 1 2010-05-01T00:00:00 TMAX     GHCND:USW0~   222 ""    ""    0     2400  celciu~
## 2 2010-05-02T00:00:00 TMAX     GHCND:USW0~   222 ""    ""    0     2400  celciu~
## 3 2010-05-03T00:00:00 TMAX     GHCND:USW0~   233 ""    ""    0     2400  celciu~
## 4 2010-05-04T00:00:00 TMAX     GHCND:USW0~   222 ""    ""    0     2400  celciu~
## 5 2010-05-05T00:00:00 TMAX     GHCND:USW0~   272 ""    ""    0     2400  celciu~
## 6 2010-05-06T00:00:00 TMAX     GHCND:USW0~   194 ""    ""    0     2400  celciu~
```

```
ncdc_plot(with_units, breaks="45 days")
```



1.3 Evaluating Records

TBD

1.4 Export Options

TBD

2 Sea Surface Temperature Data – SURP PROJECT WAITING TO HAPPEN

In contrast to terrestrial data, sea surface temperature (SST) is quite difficult to obtain and process. There are numerous tools to access the data, but they often require knowledge of complex software tools that are not easy to set up or programming experience with python or others.

`https://climexp.knmi.nl/select.cgi?id=someone@somewhere&field=ersstv5`

There are, however, a few tools build for R users that seem to accomplish all that we need.

`https://rda.ucar.edu/index.html?hash=data_user&action=register`

`https://rda.ucar.edu/datasets/ds277.9/`

Alternatively, we can download flat ascII tables of gridded data:

`https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/`

```
library(chron)
library(RColorBrewer)
library(lattice)
#library(ncdf)
library(ncdf4)
#library(greenbrown) # for gridded trend analysis

ersst.nc = "/home/CAMPUS/mwl04747/github/Climate_Change_Narratives/Data/FA19/ersst.v5.185401
Y1854 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1854.asc"
Y1864 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1864.asc"
Y1874 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1874.asc"
Y1884 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1884.asc"
Y1894 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1894.asc"
Y1904 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1904.asc"
Y1914 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1914.asc"
Y1924 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1924.asc"
Y1934 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1934.asc"
Y1944 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1944.asc"
Y1954 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1954.asc"
Y1964 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1964.asc"
Y1974 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1974.asc"
Y1984 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1984.asc"
Y1994 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.1994.asc"
Y2004 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.2004.asc"
Y2014 = "https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/ersst.v5.2014.asc"
```

```r
temp = rbind(read.table(Y1854)[75,67], read.table(Y1864)[75,67], read.table(Y1874)[75,67],
read.table(Y1884)[75,67], read.table(Y1894)[75,67], read.table(Y1904)[75,67],
read.table(Y1914)[75,67], read.table(Y1924)[75,67], read.table(Y1934)[75,67],
read.table(Y1944)[75,67], read.table(Y1954)[75,67], read.table(Y1964)[75,67],
read.table(Y1974)[75,67], read.table(Y1984)[75,67], read.table(Y1994)[75,67],
read.table(Y2004)[75,67], read.table(Y2014)[75,67])

temp.df = data.frame(Temp = as.vector(temp)/100); temp.df
temp.df$Year = seq(1854, 2014, 10)
plot(Temp~ Year, temp.df)
abline(coef(lm(Temp~Year, data=temp.df)), col="red")
#automating this process!

directory = "/pub/data/cmb/ersst/v5/ascii"

B195401 = nc_open(ersst.nc)


# str(B195401)
# print(B195401)

ncin = B195401

print(ncin)
lon <- ncvar_get(ncin, "lon")
nlon <- dim(lon)
head(lon)

lat <- ncvar_get(ncin, "lat", verbose = F)
nlat <- dim(lat)
head(lat)

print(c(nlon, nlat))

t <- ncvar_get(ncin, "time")
tunits <- ncatt_get(ncin, "time", "units")
nt <- dim(t); nt

lat.sel = 67; lon.set = 75

#ncvar_get(ncin, sst) #object 'sst' not found

#ncvar_get(ncin, var£sst) object of type 'closure' is not subsettable
#ncvar_get(ncin, var) second argument to ncvar_get must be an object of type ncvar or ncdim
```

```r
ncvar_get(ncin, "sst") #spits out the temperatures. but why the negative numbers!

# tmp.array <- ncvar_get(ncin, dname) # doesn't work...

tmp.array <- ncvar_get(ncin, "sst")
dim(tmp.array)

tmp.array[75, 67]

tmp.array[67,]

dlname <- ncatt_get(ncin, "sst", "long_name")
dunits <- ncatt_get(ncin, "sst", "units")
fillvalue <- ncatt_get(ncin, "sst", "_FillValue")
dim(tmp.array)

title <- ncatt_get(ncin, 0, "title")
institution <- ncatt_get(ncin, 0, "institution")
datasource <- ncatt_get(ncin, 0, "source")
references <- ncatt_get(ncin, 0, "references")
history <- ncatt_get(ncin, 0, "history")
Conventions <- ncatt_get(ncin, 0, "Conventions")

# split the time units string into fields
tustr <- strsplit(tunits$value, " ")
tdstr <- strsplit(unlist(tustr)[3], "-")
tmonth = as.integer(unlist(tdstr)[2])
tday = as.integer(unlist(tdstr)[3])
tyear = as.integer(unlist(tdstr)[1])
chron(t, origin = c(tmonth, tday, tyear))

# tmp.array[tmp.array == fillvalue$value] <- NA

# length(na.omit(as.vector(tmp.array[, , 1])))

m <- 1
tmp.slice <- tmp.array[, , m]

image(lon, lat, tmp.array, col = rev(brewer.pal(10, "RdBu")))

# image(lon, lat, tmp.slice, col = rev(brewer.pal(10, "RdBu")))
```

3 Satellite Data

TBD

4 Ice-Core Data

TBD