LST-R-Raster

Aji John(ajijohn@uw.edu

8/28/2017

Load the required

```
library(sp)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

R Markdown

Get the latest imagery, version with Lat/Long File downloaded - http://thredds.ucar.edu/thredds/ncss/satellite/SFC-T/SUPER-NATIONAL_1km/current/SUPER-NATIONAL_1km_SFC-T_20170829_1600.gini/dataset.html (by choosing Add 2D Lat/Lon to file (if needed for CF compliance) - Add Lat/Lon variables on the lower right)

```
#link to the latest satellite imagery(downloaded)
latest <-'SUPER-NATIONAL_1km_SFC-T_20170829_1600.gini.nc4'

#or
#link to the latest satellite imagery
#latest <-'http://thredds.ucar.edu/thredds/dodsC/satellite/SFC-T/SUPER-NATIONAL_1km/current/SUPER-NATIO
```

Including Plots

Open the gini as netCDF

```
# read the GINI as netCDF file
# load the ncdf4 package
library(ncdf4)
# open a netCDF file
ncin <- nc_open(latest)
print(ncin)
```

```
## File SUPER-NATIONAL_1km_SFC-T_20170829_1600.gini.nc4 (NC_FORMAT_NETCDF4):
##
##      4 variables (excluding dimension variables):
##         unsigned byte SFC_T[x,y,time]   (Chunking: [1536,170,1])  (Compression: shuffle,level 5)
##             long_name: Surface Skin Temperature
##             units: N/A
```

```
##             _Unsigned: true
##             scale_factor: 1
##             add_offset: 0
##             _CoordinateAxes: x y time
##             coordinates: time y x
##             grid_mapping: Stereographic
##         char Stereographic[]   (Contiguous storage)
##             grid_mapping_name: stereographic
##             longitude_of_projection_origin: -105
##             latitude_of_projection_origin: 90
##             scale_factor_at_projection_origin: 0.93301269409307
##             earth_radius: 6371229
##             _CoordinateTransformType: Projection
##             _CoordinateAxes: x y
##         double lat[x,y]   (Chunking: [1536,21])  (Compression: shuffle,level 5)
##             units: degrees_north
##             long_name: latitude coordinate
##             standard_name: latitude
##             _CoordinateAxisType: Lat
##         double lon[x,y]   (Chunking: [1536,21])  (Compression: shuffle,level 5)
##             units: degrees_east
##             long_name: longitude coordinate
##             standard_name: longitude
##             _CoordinateAxisType: Lon
##
##      3 dimensions:
##         time  Size:1
##             long_name: time since base date
##             _CoordinateAxisType: Time
##             units: msecs since 1970-01-01T00:00:00Z
##         y  Size:1008
##             long_name: projection y coordinate
##             units: km
##             _CoordinateAxisType: GeoY
##             standard_name: projection_y_coordinate
##         x  Size:1536
##             long_name: projection x coordinate
##             units: km
##             _CoordinateAxisType: GeoX
##             standard_name: projection_x_coordinate
##
##     33 global attributes:
##         Conventions: CF-1.0
##         source_id: 1
##         entity_id: 6
##         sector_id: 9
##         phys_elem: 18
##         time_coverage_start: 2017-08-29T16:00:00
##         time_coverage_end: 2017-08-29T16:00:00
##         ProjIndex: 5
##         ProjName: POLARSTEREOGRAPHIC
##         NX: 1536
##         NY: 1008
##         Lov: -105
```

```
##         DxKm: 7.9465
##         DyKm: 7.9465
##         ProjCenter: 1
##         Latin: 0
##         title: Composite
##         summary: Sounder Based Derived Surface Skin Temperature
##         id: Supernational
##         keywords_vocabulary: SFC_T
##         cdm_data_type: GRID
##         featureType: GRID
##         standard_name_vocabulary: Surface Skin Temperature
##         creator_name: UNIDATA
##         creator_url: http://www.unidata.ucar.edu/
##         naming_authority: UCAR/UCP
##         geospatial_lat_min: 7.78898592077309
##         geospatial_lat_max: 35.5883624882841
##         geospatial_lon_min: -141.032242028194
##         geospatial_lon_max: -18.5232659108977
##         imageResolution: 1
##         compressionFlag: 0
##         History: Translated to CF-1.0 Conventions by Netcdf-Java CDM (CFGridWriter2)
## Original Dataset = /data/ldm/pub/native/satellite/SFC-T/SUPER-NATIONAL_1km/current/SUPER-NATIONAL_1k
```

get the surface temperature variable

```
sfct <- ncvar_get(ncin, "SFC_T")
dim(sfct)
```

```
## [1] 1536 1008
```

sfct is a two dimenional array

```
# get proj x
x <- ncvar_get(ncin,"x")
nx <- dim(x)
head(nx)
```

```
## [1] 1536
```

```
summary(x)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -6096.000 -3047.000     2.907     2.907  3052.000  6102.000
```

```
# get proj y
y <- ncvar_get(ncin,"y")
ny <- dim(y)
head(ny)
```

```
## [1] 1008
```

```
summary(y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -8382.0 -6382.0 -4381.0 -4381.0 -2380.0  -379.9
```

Time is only for onetimestamp

```
print(c(nx,ny))
```

```
## [1] 1536 1008
```

```
# get time
time <- ncvar_get(ncin,"time")
time
```

```
## [1] 1.504022e+12
```

```
tunits <- ncatt_get(ncin,"time","units")
nt <- dim(time)
as.character(ncin$dim$time$vals)
```

```
## [1] "1504022400000"
```

Assuming that this timestamp is in milliseconds: GMT: Friday, August 25, 2017 7:00:00 PM

Get lat long

```
# get lat
lat <- ncvar_get(ncin,"lat")
nlat <- dim(lat)
lat_df <- data.frame(lat = as.vector(lat))
summary(lat_df)
```

```
##       lat
##  Min.   : 7.819
##  1st Qu.:26.553
##  Median :37.810
##  Mean   :39.832
##  3rd Qu.:51.164
##  Max.   :86.339
```

```
# get long
long <- ncvar_get(ncin,"lon")
long_df <- data.frame(lon = as.vector(long))
summary(long_df)
```

```
##       lon
##  Min.   :-180.00
##  1st Qu.:-136.45
##  Median :-100.15
##  Mean   : -94.09
##  3rd Qu.: -66.85
##  Max.   : 180.00
```

Our goal is to create spatial points dataframe Note:latitude is represented by horizontal lines, which go up and down (North and South) the Y axis, and Longitude is X

```
# get proj y
xy_df <- data.frame(lon=long_df$lon,lat=lat_df$lat)

coordinates(xy_df)= ~lon + lat
sfct_df <- data.frame(sfct = as.vector(sfct))
lst_sp_fw <- SpatialPointsDataFrame(coordinates(xy_df), data =sfct_df
                                    , proj4string = CRS("+proj=longlat +ellps=WGS84"))
```

Now, we created the spatial points dataframe and existing points are Sterographic , and the units are in km next, we transform it to lat/long, not needed as we have the netCDF already with lat/long

```
#lst_sp_fw_proj <- spTransform(lst_sp_fw, CRS("+proj=longlat +ellps=WGS84"))
```

Verify the bounding box

```
summary(lst_sp_fw)
```

```
## Object of class SpatialPointsDataFrame
## Coordinates:
##             min       max
## lon -179.999867 179.99991
## lat    7.819369  86.33945
## Is projected: FALSE
## proj4string : [+proj=longlat +ellps=WGS84]
## Number of points: 1548288
## Data attributes:
##       sfct
##  Min.   :  0.00
##  1st Qu.:  0.00
##  Median :  0.00
##  Mean   : 33.16
##  3rd Qu.:  0.00
##  Max.   :239.00
```

```
bbox(lst_sp_fw)
```

```
##             min       max
## lon -179.999867 179.99991
## lat    7.819369  86.33945
```

Lets try to plot it, to see if the coordinates are correct.

```
# commemt
library(ggmap)
```

```
## Loading required package: ggplot2
```

```
## Google Maps API Terms of Service: http://developers.google.com/maps/terms.
```

```
## Please cite ggmap if you use it: see citation("ggmap") for details.
```

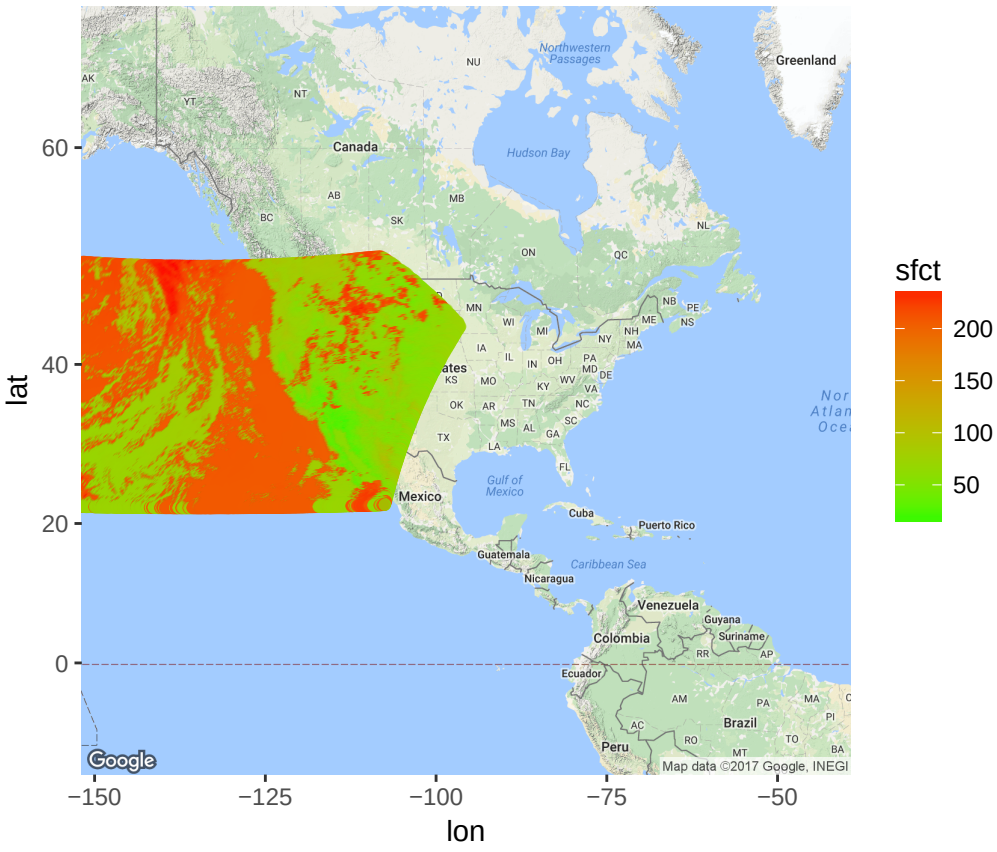
```
us_map<-get_map(location='united states', zoom=3, maptype = "terrain",
            source='google',color='color')
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=united+states&zoom=3&size=640x640&scal
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=united%20states
```

```
# commemt
lst_df_final <- as.data.frame(lst_sp_fw)
# Remove zero surface skin temperature
lst_df_final_trimmed =  lst_df_final %>% filter(sfct > 0) %>% as.data.frame()

ggmap(us_map) + geom_point(
        aes(x=lon, y=lat,  colour=sfct),
        data=lst_df_final_trimmed, alpha=.5, na.rm = T)  +
        scale_color_gradient(low="green", high="red")
```


lat
60
40
20
0
−150
−125
−100
−75
−50
lon
sfct
200
150
100
50
Greenland
Canada
Hudson Bay
Northwestern Passages
Mexico
Gulf of Mexico
Cuba
Puerto Rico
Guatemala
Nicaragua
Caribbean Sea
Venezuela
Guyana
Suriname
Colombia
Ecuador
Brazil
Peru
Google
Map data ©2017 Google, INEGI