

Download and plot PRISM data

Edmund Hart

2015-11-11

This package allows users to access and visualize data from the [Oregon State PRISM project](#). Data is all in the form of gridded rasters for the continental US at 3 different scales: daily, monthly and 30 year normals. Please see their webpage for a full description of the data products, or [see their overview](#).

Downloading data

Data is available in 3 different forms as mentioned above. Each one has it's own function to download data. While each data type has slightly different temporal parameters, the type options are always the same. Keep in mind these are modeled parameters, not measured. Please see the [full description](#) for how they are calculated.

Parameter name	Description
<i>tmean</i>	Mean temperature
<i>tmax</i>	Maximum temperature
<i>tmin</i>	Minimum temperature
<i>ppt</i>	Total precipitation (Rain and snow)
<i>vpdmin</i>	Daily minimum vapor pressure deficit [averaged over all days in the month - normal data only]
<i>vpdmax</i>	Daily maximum vapor pressure deficit [averaged over all days in the month - normal data only]

Normal Data

Normals are based on the years 1981 - 2010, and can be downloaded in two resolutions, 4km and 800m, and a resolution must be specified. Normals can also be downloaded for a given month, vector of months, or an average for all 30 years.

```
library(prism)
options(prism.path = "~/prismtmp")
get_prism_normals(type="tmean",resolution = "4km",mon = 1:6, keepZip=F)
```

The first thing to note is that you'll need to set a local location to work with this data. Second is the option `keepZip`. If this is `TRUE` the zip file will remain on your machine, otherwise it will be automatically deleted.

You can also view all the data you have downloaded with a simple command `ls_prism_data()`. By default this just gives a list of file names. All the internal functions in the package work off of this simple list of files.

```
## Truncated to keep file list short
ls_prism_data()[1:10,]

## [1] "PRISM_tmean_30yr_normal_4kmM2_01_bil"
## [2] "PRISM_tmean_30yr_normal_4kmM2_02_bil"
## [3] "PRISM_tmean_30yr_normal_4kmM2_03_bil"
## [4] "PRISM_tmean_30yr_normal_4kmM2_04_bil"
## [5] "PRISM_tmean_30yr_normal_4kmM2_05_bil"
## [6] "PRISM_tmean_30yr_normal_4kmM2_06_bil"
## [7] "PRISM_tmean_30yr_normal_4kmM2_annual_bil"
```

```
## [8] "PRISM_tmean_stable_4kmD1_20130601_bil"
## [9] "PRISM_tmean_stable_4kmD1_20130602_bil"
## [10] "PRISM_tmean_stable_4kmD1_20130603_bil"
```

While internal plotting functions use this, other files may want an absolute path (e.g. the **raster** package), there's a parameter **absPath** that conveniently returns the absolute path. Alternatively you may want to see what the normal name for the product is (not the file name), and that parameter is **name**.

```
ls_prism_data(absPath = TRUE)[1:10,]
```

```
##                                files
## 1    PRISM_tmean_30yr_normal_4kmM2_01_bil
## 2    PRISM_tmean_30yr_normal_4kmM2_02_bil
## 3    PRISM_tmean_30yr_normal_4kmM2_03_bil
## 4    PRISM_tmean_30yr_normal_4kmM2_04_bil
## 5    PRISM_tmean_30yr_normal_4kmM2_05_bil
## 6    PRISM_tmean_30yr_normal_4kmM2_06_bil
## 7    PRISM_tmean_30yr_normal_4kmM2_annual_bil
## 8    PRISM_tmean_stable_4kmD1_20130601_bil
## 9    PRISM_tmean_stable_4kmD1_20130602_bil
## 10   PRISM_tmean_stable_4kmD1_20130603_bil
##
##                                abs_path
## 1    ~/prismtmp/PRISM_tmean_30yr_normal_4kmM2_01_bil/PRISM_tmean_30yr_normal_4kmM2_01_bil.bil
## 2    ~/prismtmp/PRISM_tmean_30yr_normal_4kmM2_02_bil/PRISM_tmean_30yr_normal_4kmM2_02_bil.bil
## 3    ~/prismtmp/PRISM_tmean_30yr_normal_4kmM2_03_bil/PRISM_tmean_30yr_normal_4kmM2_03_bil.bil
## 4    ~/prismtmp/PRISM_tmean_30yr_normal_4kmM2_04_bil/PRISM_tmean_30yr_normal_4kmM2_04_bil.bil
## 5    ~/prismtmp/PRISM_tmean_30yr_normal_4kmM2_05_bil/PRISM_tmean_30yr_normal_4kmM2_05_bil.bil
## 6    ~/prismtmp/PRISM_tmean_30yr_normal_4kmM2_06_bil/PRISM_tmean_30yr_normal_4kmM2_06_bil.bil
## 7    ~/prismtmp/PRISM_tmean_30yr_normal_4kmM2_annual_bil/PRISM_tmean_30yr_normal_4kmM2_annual_bil.bil
## 8    ~/prismtmp/PRISM_tmean_stable_4kmD1_20130601_bil/PRISM_tmean_stable_4kmD1_20130601_bil.bil
## 9    ~/prismtmp/PRISM_tmean_stable_4kmD1_20130602_bil/PRISM_tmean_stable_4kmD1_20130602_bil.bil
## 10   ~/prismtmp/PRISM_tmean_stable_4kmD1_20130603_bil/PRISM_tmean_stable_4kmD1_20130603_bil.bil
```

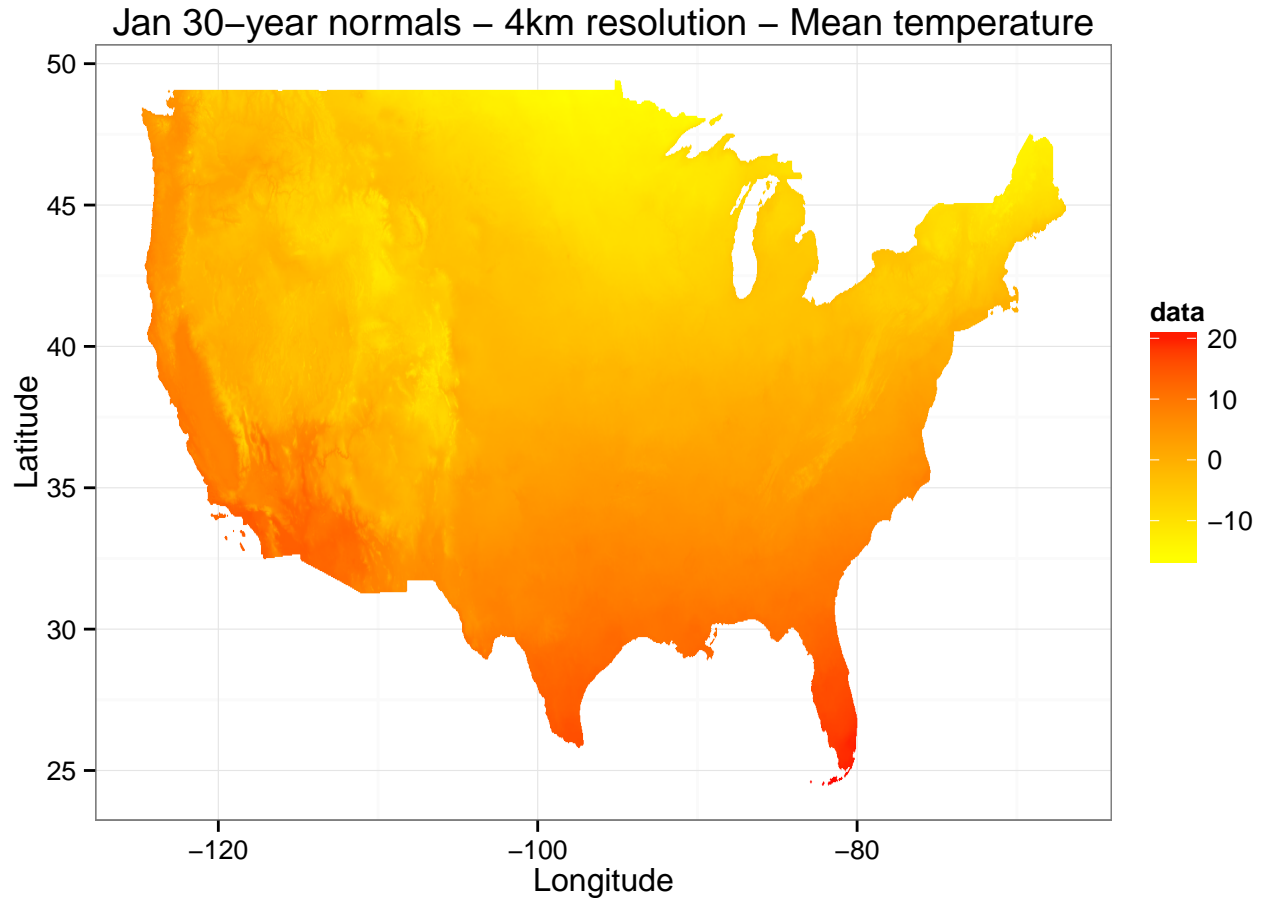
```
ls_prism_data(name = TRUE)[1:10,]
```

```
##                                files
## 1    PRISM_tmean_30yr_normal_4kmM2_01_bil
## 2    PRISM_tmean_30yr_normal_4kmM2_02_bil
## 3    PRISM_tmean_30yr_normal_4kmM2_03_bil
## 4    PRISM_tmean_30yr_normal_4kmM2_04_bil
## 5    PRISM_tmean_30yr_normal_4kmM2_05_bil
## 6    PRISM_tmean_30yr_normal_4kmM2_06_bil
## 7    PRISM_tmean_30yr_normal_4kmM2_annual_bil
## 8    PRISM_tmean_stable_4kmD1_20130601_bil
## 9    PRISM_tmean_stable_4kmD1_20130602_bil
## 10   PRISM_tmean_stable_4kmD1_20130603_bil
##
##                                product_name
## 1    Jan 30-year normals - 4km resolution - Mean temperature
## 2    Feb 30-year normals - 4km resolution - Mean temperature
## 3    Mar 30-year normals - 4km resolution - Mean temperature
## 4    Apr 30-year normals - 4km resolution - Mean temperature
## 5    May 30-year normals - 4km resolution - Mean temperature
```

```
## 6      Jun 30-year normals - 4km resolution - Mean temperature
## 7  Annual 30-year normals - 4km resolution - Mean temperature
## 8      Jun 01 2013 - 4km resolution - Mean temperature
## 9      Jun 02 2013 - 4km resolution - Mean temperature
## 10     Jun 03 2013 - 4km resolution - Mean temperature
```

You can easily make a quick plot of your data to using the output of `ls_prism_data()`

```
prism_image(ls_prism_data()[1,1])
```



Monthly and Daily Data

Monthly and daily data is also easily accessible. Below we'll get January data for the years 1990 to 2000. We also grab data from June 1 to June 14 2013.

```
get_prism_monthlys(type="tmean", year = 1990:2000, mon = 1, keepZip=F)
get_prism_dailys(type="tmean", minDate = "2013-06-01", maxDate = "2013-06-14", keepZip=F)
```

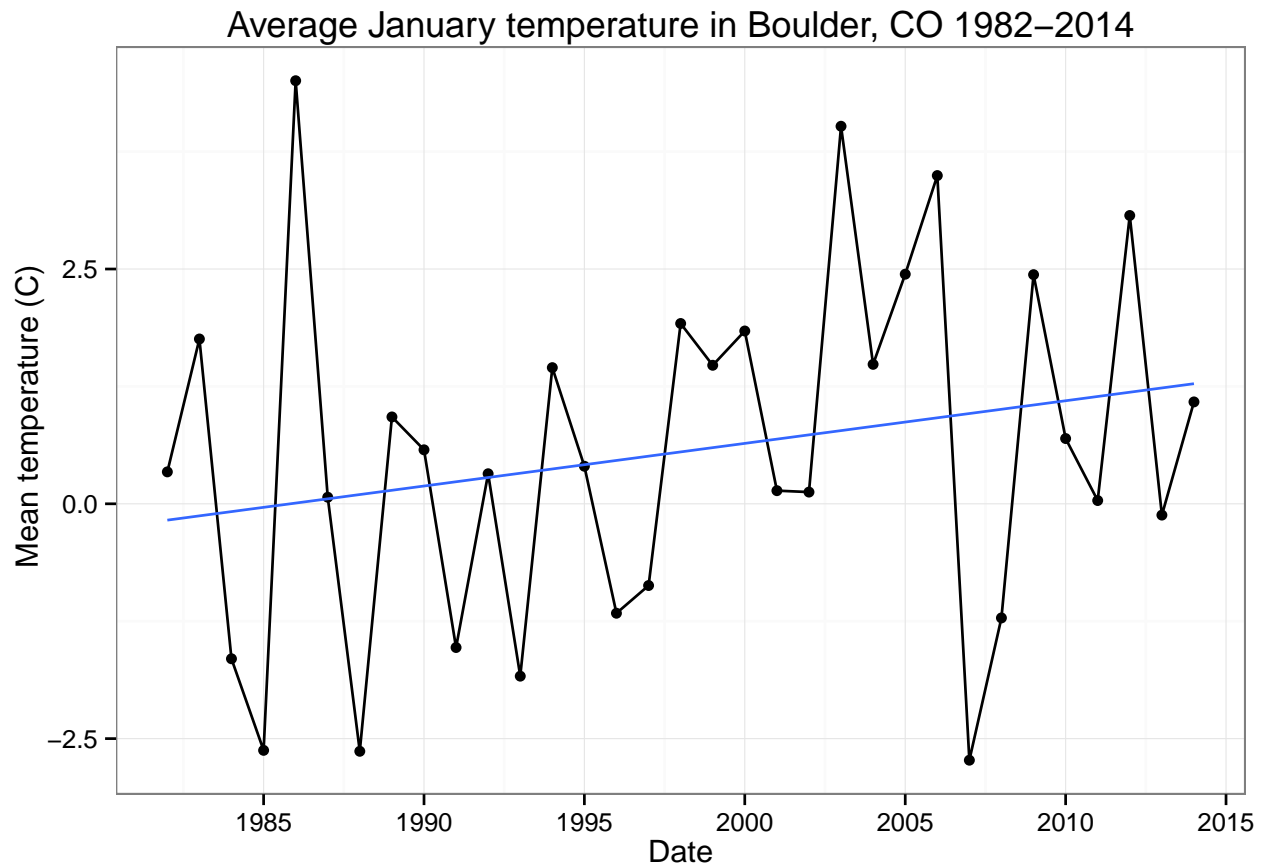
Note that for daily data you need to give a well formed date string in the form of "YYYY-MM-DD"

You can also visualize a single point across a set of rasters. This procedure will take a set of rasters, create a stack, extract data at a point, and then create a ggplot2 object.

Let's get make a plot of January temperatures in Boulder between 1982 and 2014. First we'll grab all the data from the US, and then give our function a point to get data from. The point must be a vector in the form of longitude, latitude.

```
## Get data.
get_prism_monthlys(type="tmean", year = 1982:2014, mon = 1, keepZip=F)
```

```
library(ggplot2)
boulder <- c(-105.2797,40.0176)
## We'll use regular expressions to grep through the list and get data only from
## the month of January
to_slice <- grep("_[0-9]{4}[0][1]",ls_prism_data()[,1],value=T)
to_slice = grep("tmean",to_slice, value = T)
p <- prism_slice(boulder,to_slice)
p + stat_smooth(method="lm",se=F) + theme_bw() +
ggtitle("Average January temperature in Boulder, CO 1982-2014")
```



Lastly it's easy to just load up the prism data with the raster package. This time what we'll look at January temperature anomalies. To do this we'll examine the difference between January 2013 and the 30 year normals for January. Conveniently, we've already downloaded both of these files. We just need to grab them out of our list.

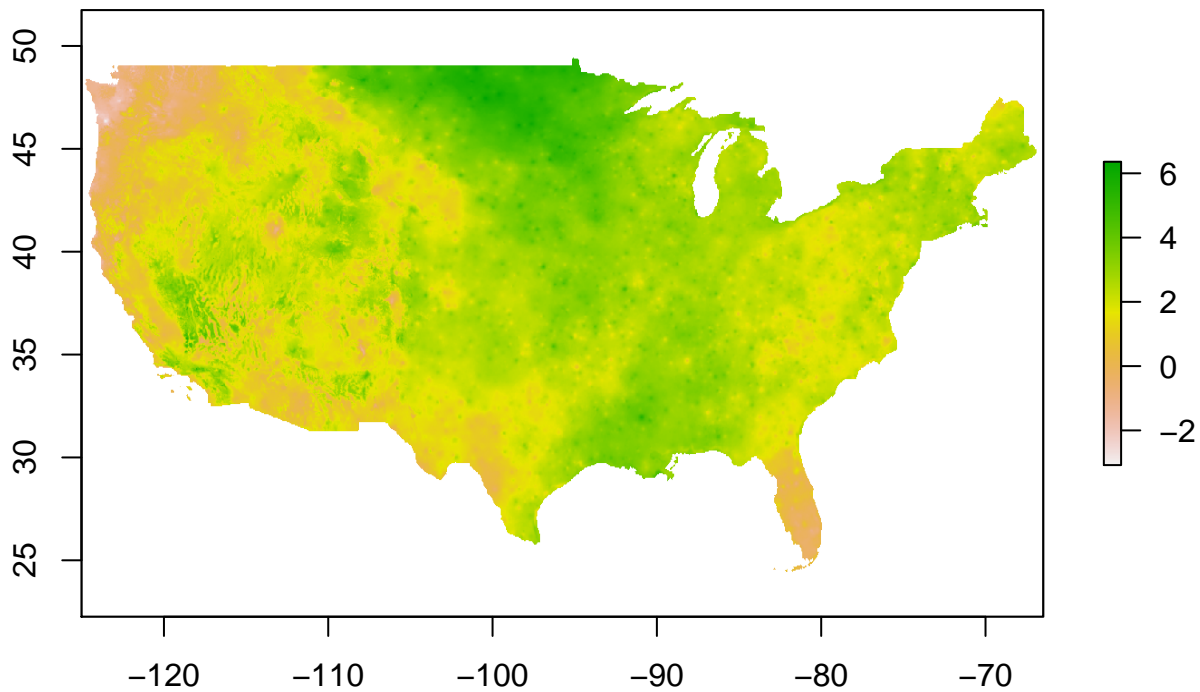
```
library(raster)
### I got these just by looking at the list output
jnrm <- ls_prism_data(absPath=T)[1,2]
j2013 <- ls_prism_data(absPath=T)[52,2]
## See that the full path is returned
jnrm
```

```
## [1] "~/prismtmp/PRISM_tmean_30yr_normal_4kmM2_01_bil/PRISM_tmean_30yr_normal_4kmM2_01_bil.bil"
```

```
## Now we'll load the rasters.
jnorm_rast <- raster(jnorm)
j2013_rast <- raster(j2013)
## Now we can do simple subtraction to get the anomaly by subtracting 2014 from
## the 30 year normal map
anomCalc <- function(x, y) {
  return(x - y)
}

anom_rast <- overlay(j2013_rast,jnorm_rast,fun = anomCalc)

plot(anom_rast)
```



The plot shows that January 2013 was warmer than the average over the last 30 years. It also shows how easy it is to use the raster library to work with prism data. The package provides a simple framework to work with a large number of rasters that you can easily download and visualize or use with other data sets.