Data Inventory

Ben DeVries

April 17, 2014

A number of functions are available in the bfastSpatial package to help keep an inventory of data in a raster time series stack. These functions range from basic scene information (`getSceneinfo()`) to summary of pixel values per year in the time series (`annualSummary()`).

The following functions are included in the Data Inventory module:

1. `getSceneinfo()`
2. `countObs()`
3. `annualSummary()`

1 Basic Scene Information: `getSceneinfo()`

`getSceneinfo()` allows the user to list the information contained within a scene ID. Currently, only Landsat scene ID's are supported. For example, the scene ID "LE71700552007309ASN00" tells us that the scene is from the Landsat 7 ETM+ sensor ('LE7'), path-row 170-55 ('170055') and was acquired on the 309th day of the year 2007 ('2007309'). Calling `getSceneinfo('LE71700552007309ASN00')` will give a data.frame with one row showing all of this information.

```
## Loading required package: raster
## Loading required package: sp
## Loading required package: parallel
## Loading required package: bfast
## Loading required package: strucchange
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##    as.Date, as.Date.numeric
##
## Loading required package: sandwich
```

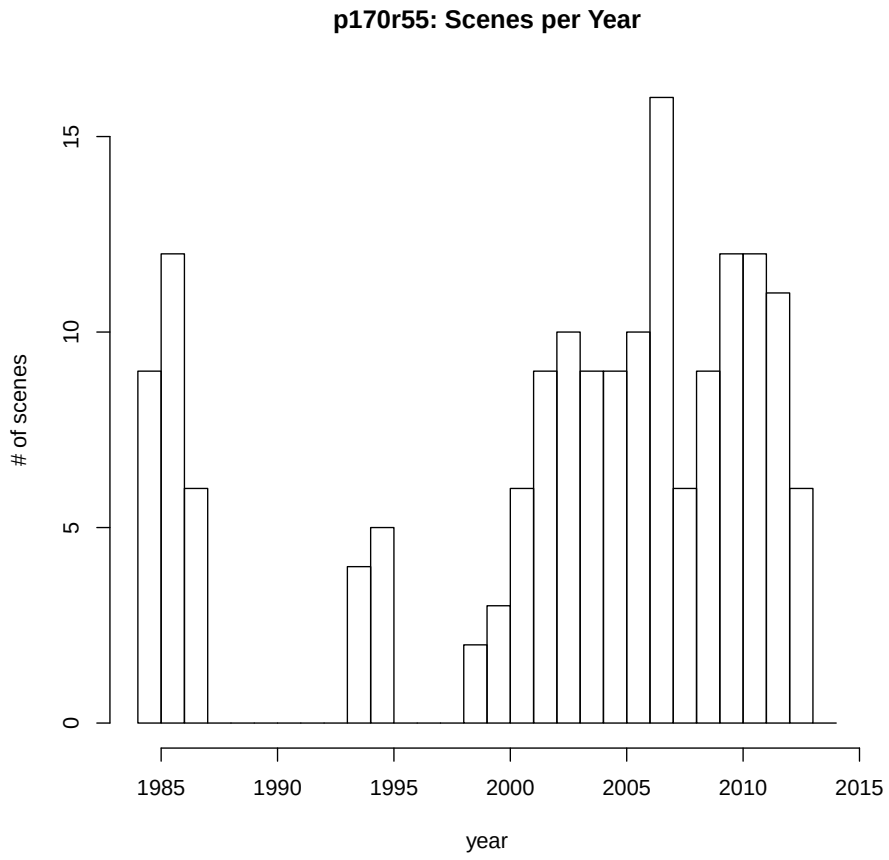
```
## Loading required package:  MASS
##
## Attaching package:  'MASS'
##
## The following objects are masked from 'package:raster':
##
##    area, select
##
## Loading required package:  forecast
## Loading required package:  timeDate
## This is forecast 5.3
##
## Loading required package:  gdalUtils
## Loading required package:  stringr
## Warning:  replacing previous import by 'raster::edge' when loading
'bfastSpatial'
## Warning:  replacing previous import by 'raster::edges' when loading
'bfastSpatial'
```

```
getSceneinfo("LE71700552007309ASN00")

##                         sensor path row       date
## LE71700552007309 ETM+ SLC-off  170  55 2007-11-05
```

When working with Landsat data, it is a good idea to assign and keep these sceneID's as layer names (see `?raster::names`) so the relevant information is associated to each raster layer.

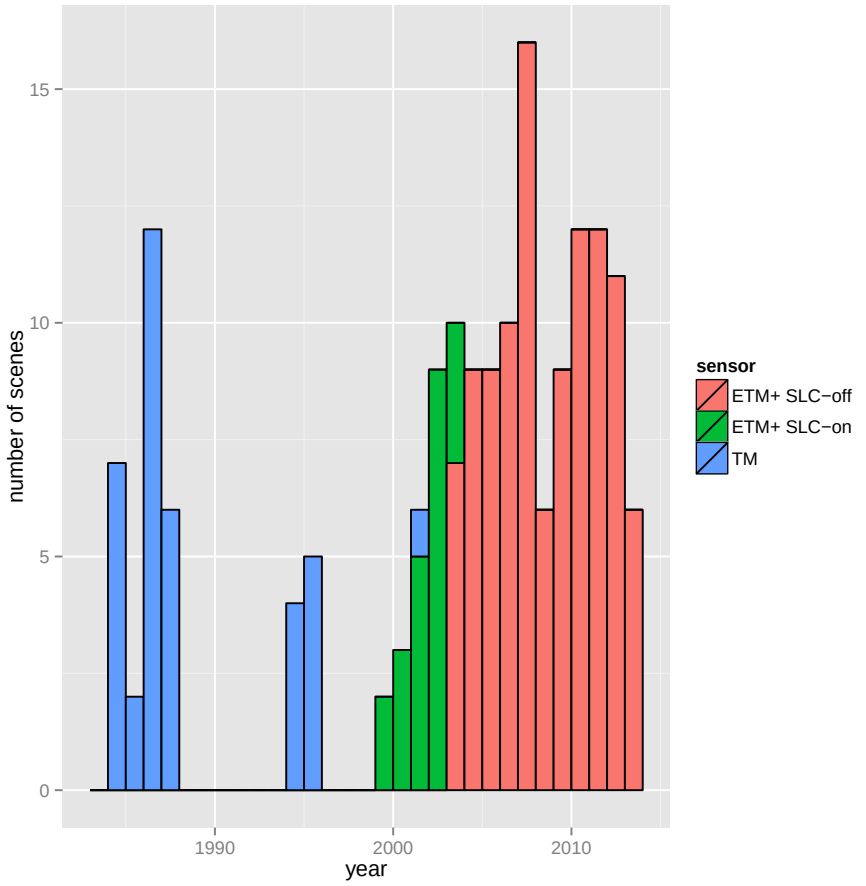
```
# show scene info from tura layers
data(tura)
head(names(tura))

## [1] "LE71700551999255AGS00" "LE71700551999271EDC00" "LE71700552000114SGS00"
## [4] "LE71700552000194EDC00" "LE71700552000258SGS00" "LE71700552001036SGS00"

s <- getSceneinfo(names(tura))
head(s)

##                        sensor path row       date
## LE71700551999255 ETM+ SLC-on  170  55 1999-09-12
## LE71700551999271 ETM+ SLC-on  170  55 1999-09-28
## LE71700552000114 ETM+ SLC-on  170  55 2000-04-23
## LE71700552000194 ETM+ SLC-on  170  55 2000-07-12
## LE71700552000258 ETM+ SLC-on  170  55 2000-09-14
## LE71700552001036 ETM+ SLC-on  170  55 2001-02-05
```

```
# add a column for years and plot # of scenes per year
s$year <- as.numeric(substr(s$date, 1, 4))
hist(s$year, breaks = c(1984:2014), main = "p170r55: Scenes per Year", xlab = "year",
    ylab = "# of scenes")
```



We can combine the dates and sensor information to get more of an idea of where our data are coming from.

```
library(ggplot2)
p <- ggplot(data = s, aes(x = year))
p <- p + geom_bar(aes(fill = sensor), binwidth = 1, col = "black")
p <- p + labs(y = "number of scenes")
p
```



More examples can be found under `?getSceneinfo`. Many other functions in the bfastSpatial package rely on getSceneinfo to extract relevant scene information, such as acquisition dates to be passed to `bfmSpatial()` or `bfmPixel()`.

2 Valid Observations: `countObs()`

The number of available observations in a raster time series can be calculated by using countObs(). This function "drills" through pixels of a time series and counts the number of pixels with a non-NA value. Optionally, any other value can be supplied as a substitute for NA (e.g. the number of non-zero values per pixel can also be queried). Values can also be expressed as a percentage if as.perc is set to TRUE.

```
data(tura)
obs <- countObs(tura, as.perc = TRUE)

## Error:  could not find function "countObs"
```

```
plot(obs)
```

```
## Error:  error in evaluating the argument 'x' in selecting a method
for function 'plot':  Error:  object 'obs' not found
```

```
summary(obs)
```

```
## Error:  error in evaluating the argument 'object' in selecting a
method for function 'summary':  Error:  object 'obs' not found
```

3 Annual Summary Statistics: `annualSummary()`