

Useful.Pacakages.For.Shiny

Matthew Ross

November 11, 2015

This is an RMarkdown version of the Useful.Packags.For.Shiny.Apps.R script. Hopefully it shows how R Markdown is a great way to structure code that will be shared or requires lots of commenting. Let's get started.

Packages

R is all about packages. Finding, using, and understanding the best ones can make coding incredibly efficient and fast. These are my favorite ones for Shiny apps. From here on I'll highlight the package names and then after will be a First we need to get some data to use as an example using a great package from the USGS `##EGRET ##dataRetrieval`

Reads in data from the Mud River Reservoir in West Virginia that drains my research catchments. USGS code '03204250'

```
library(EGRET)
library(dataRetrieval)
#First number is USGS station id. Second number is data code for water height.
z <- readNWISDaily('03204250','00065')
```

```
## There are 4384 data points, and 4429 days.
##
## discharge data jumps from 2009-05-10 to 2009-05-15
## discharge data jumps from 2009-07-21 to 2009-07-25
## discharge data jumps from 2009-07-29 to 2009-08-04
## discharge data jumps from 2009-08-16 to 2009-08-18
## discharge data jumps from 2011-03-02 to 2011-03-06
## discharge data jumps from 2011-03-07 to 2011-03-12
## discharge data jumps from 2011-03-12 to 2011-03-23
## discharge data jumps from 2011-06-05 to 2011-06-10
## discharge data jumps from 2013-03-06 to 2013-03-12
## discharge data jumps from 2013-03-22 to 2013-03-27
## discharge data jumps from 2015-01-25 to 2015-01-28
```

Oooh, look at all those warnings! For now we will keep them, but there are ways to supress them if you want a cleaner document. Now that we have data we need to convert the timestamps into a useful format using a package from the great Hadley Wickham.

lubridate

```
library(lubridate)
#Convert timestamp to a format that R can read.
#Lubridate can handle all kinds of formats but in this case it is in year month day. TZ is for timezone
z$lubridate <- ymd(z$date,tz='Etc/GMT-5')
```

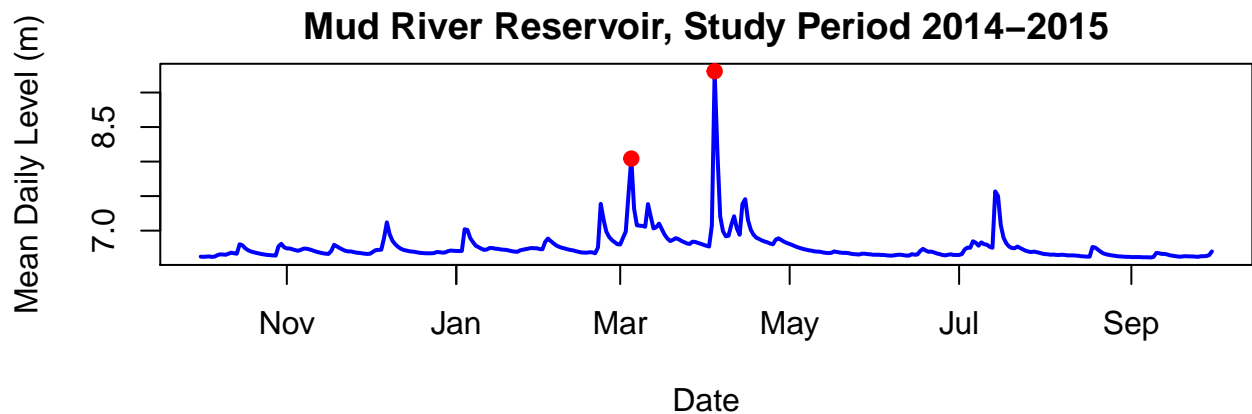
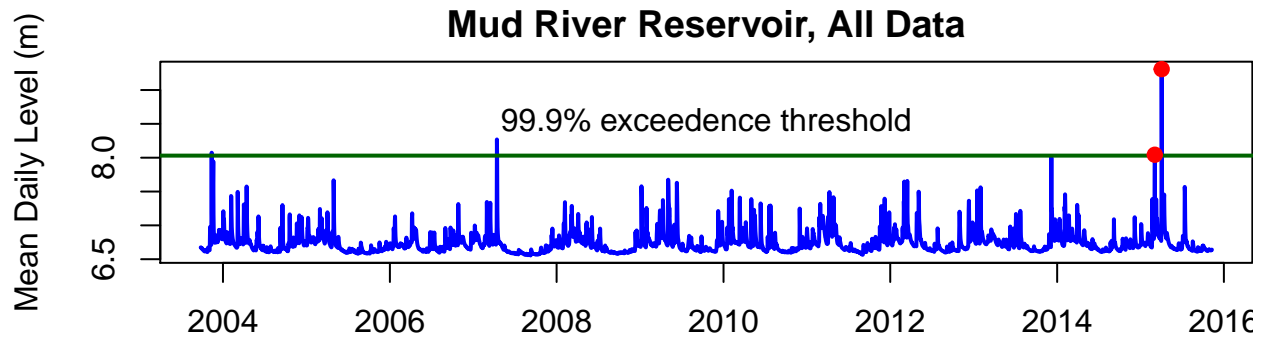
```

#Convert level data into meters
z$meter <- z$Q*.3048
#Setup plot parameters
par(mfrow=c(2,1),cex=1,mar=c(4,4,2,0))
#Highlight days we sampled water chemistry.
sample.dates <- mdy(c('3/5/15','4/4/15'),tz='Etc/GMT-5')
#Subset sampled days from full dataset
smpl <- z[z$lubridate %in% sample.dates,]
#Plot data
plot(z$lubridate,z$meter, col='blue',lwd=2,type='l',ylab="Mean Daily Level (m)",main='Mud River Reservoir')
#Get 99.9% quantile
q.99.9 <- quantile(z$meter,probs=0.999)
#Add bar to see events above quantile
abline(h=q.99.9,col='darkgreen',lwd=2)
#Add explanation of line
mtext('99.9% exceedence threshold',line=-2)
#Add points at times where we sampled water chemistry
points(smpl$lubridate,smpl$meter, col='red',pch=19,cex=1)

#Lubridate has a great way to get sections of time series data.
#First setup the interval of time you want.
dat.interval <- interval(mdy('10/1/2014',tz='Etc/GMT-5'),mdy('9/30/2015',tz='Etc/GMT-5'))
#Then subset your data
year2015 <- z[z$lubridate %within% dat.interval,]

#Plot subsetting data
plot(meter~lubridate,data=year2015,col='blue',lwd=2,type='l',ylab="Mean Daily Level (m)",main='Mud River Reservoir')
#Add sample points.
points(smpl$lubridate,smpl$meter, col='red',pch=19,cex=1)

```



So now we have a baseplot that highlights water level at the mud river reservoir, and this data has some nice timestamps on the x-axis and can be broken down into intervals with `lubridate`. but it is kind of static and ugly, so maybe we can make it better with other packages.

`xts`

`dygraphs`

```
library(xts) # A library to deal with time series data.
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric
```

```
library(dygraphs) # A library to plot time series data
```

```
#First I need to create a column in the full dataset that holds only the days we sampled
z$Smpl.m <- NA
z[z$lubridate %in% sample.dates, 'Smpl.m'] <- z[z$lubridate %in% sample.dates, 'meter']
```

```
# Second turn the data into an xts object that holds both the full dataset and the two dates we sampled  
# Order.by is where you put in the time series data  
z.xts <- xts(cbind(MudRiverRes_m = z$meter, Sample_Days = z$Smpl.m), order.by=z$lubridate)  
#dygraph(z.xts)
```

Ok so there is an interactive graph of our data, but it also is kind of ugly and doesn't have all the data we want to see. Time for another package. This is one of my favorites called "magrittr" which is a play on words on René Magritte who has a famous painting with the label "ceci n'est pas une pipe." Here it is.



So magrittr is all about pipes!