Documentation: rollingAvg

Austin Putz

August 24, 2016

Contents

1 Introduction 1

2 Function Inputs 1

3 How it works 1

4 Example 1

1 Introduction

This rolling average function (rollingAvg) will calculate the average of a numeric variable along a date timeline. Most rolling average functions only use a vector of numbers to get a rolling average. So this is not a typical rolling average, but uses the date column instead.

2 Function Inputs

1. data = Character: Dataset you want to get the rolling average on, data.frame class
2. value.col = Column name of the numeric variable you want to take the average of.
3. date.col = Character Column name of the date column you want to use to average over.
4. ndays = Numeric: Number of days you want to stretch from the date itself. On the first date, it will split the number of days in half so it won't be the full range.
5. verbose = Logical: Do you want to see the output as it goes, only do for small examples..

3 How it works

Starting from the first date in the date column. It will take the number of days and split it in half. For example, for `ndays` = 30, then it will reach 15 days previous from the starting date and 15 days into the future. Then this subset dataset will be averaged for that first day. It will then continue for the day after, not the next available date in the dataset. It will do all the days between the oldest and newest date in that column.

4 Example

The code will be posted in a separate file as `rollingAvg_example.R`.

First remove everything if you need to so nothing gets overwritten.

```
> # remove everything before starting
>   rm(list=ls())
>
```

First set the location where you saved the function and source it.

```
> # Where is it saved????
>   location_saved <- "~/Documents/Programming/R/Functions/rollingAvg.R"
> # load the function
>   source(location_saved)
> # remove location
>   rm(location_saved)
>
```

Load the ggplot2 library and install it if you need to with `install.packages("ggplot2")`.

```
> # load libraries
>   library(ggplot2)
>
```

The data will have two columns. One with a date from January 1, 2016 to December 31, 2016 (366 days). Each date will only have one variable sampled from 4 different distributions. The follow is how they were sampled:

1. 1st 100: Mean = 10, sd = 3
2. 2nd 100: Mean = 5, sd = 2
3. 3rd 100: Mean = 11, sd=3
4. Last 66: Mean = 15, sd=3

Numeric numbers were rounded to whole numbers.

```
> # vector of dates
>   vec_date <- seq.Date(as.Date("2016-01-01"), as.Date("2016-12-31"), by="day")
> # set seed for random number generator
>   set.seed(1234)
> # figure out breaks
>   vec1 <- round(rnorm(100, mean=10, sd=3), 0)
>   vec2 <- round(rnorm(100, mean=5,  sd=2), 0)
>   vec3 <- round(rnorm(100, mean=11, sd=3), 0)
>   vec4 <- round(rnorm(length(vec_date)-length(vec1)-length(vec2)-length(vec3), mean=15, sd
> # combine
>   vec_numbers <- c(vec1, vec2, vec3, vec4)
> # remove component vectors
>   rm(list=c("vec1", "vec2", "vec3", "vec4"))
> # combine dataset into a data frame
>   data <- data.frame(Date=vec_date, Value=vec_numbers)
> # remove vectors
>   rm(list=c("vec_date", "vec_numbers"))
>
```

Here is the first 6 lines of the data.

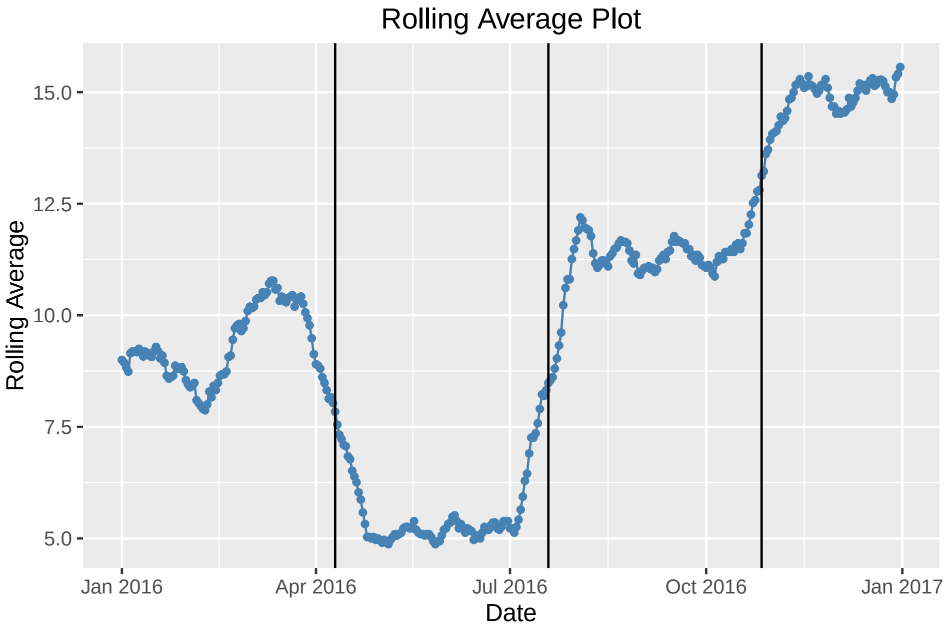
```
> # print the first lines
>   print(head(data))

        Date Value
1 2016-01-01     6
2 2016-01-02    11
3 2016-01-03    13
4 2016-01-04     3
5 2016-01-05    11
6 2016-01-06    12

>
```

The function can be called with the following code:

```
> # do rolling average
>   data.rollingAvg <- rollingAvg(data, value.col="Value", date.col="Date")
>
```

The following is a plot of the rolling averages. The vertical cuts are where the sampling changes.

```
> # plot it
>   ggplot(data.rollingAvg, aes(x=vec.dates, y=vec.means)) +
+   geom_point(shape=16, color="steelblue") +
+   geom_line(color="steelblue") +
+   geom_vline(xintercept=julian(as.Date("2016-04-10"))) +
+   geom_vline(xintercept=julian(as.Date("2016-07-19"))) +
+   geom_vline(xintercept=julian(as.Date("2016-10-27"))) +
+   ggtitle("Rolling Average Plot") +
+   xlab("Date") +
+   ylab("Rolling Average")
>
```


Rolling Average Plot
Rolling Average
Jan 2016
Apr 2016
Jul 2016
Oct 2016
Jan 2017
Date