Explore Stock Market Data

Victoria Lim

February 24, 2018

NOTE: The function documentation gets converted to markdown text, instead of being preserved as documentation. Note to self to figure out how to fix. Also need to get `stat_smooth()` to work.

```
library(stringr)    # for str_extract
library(reshape2)   # for reshaping dataframe
library(ggplot2)    # for nice plots
library(splines)    # for smoothing in plots

### directory management
#setwd("/home/victoria/Desktop/stock-market")
setwd("/home/limvt/Documents/stock-market")
```

Load a single stock file.

The data from a single stock CSV file (.txt) is loaded into a dataframe and returned. This function also extracts the stock symbol from the filename and includes it as a new column. @param filename name of stock file to read. Should be a .txt file with comma-separated values inside. @return Dataframe with the file's information. @examples load_one('aadr.us.txt') load_one('data_files/aadr.us.txt') load_one('/home/limvt/Documents/stock-market/data_files/aadr.us.txt') @export

```
load_one <- function(filename){
  # extract stock symbol from filename, before 1st . character
  filesym <- str_extract(basename(filename), "(.*?)(?=\\.)")
  d1 <- read.csv(filename)
  # convert date column to date objects
  d1$Date <- as.Date(d1$Date)
  # create a new column for stock symbol
  d1$Symbol <- filesym
  return(d1)
}


### check out 1-2 stocks
d1 <- load_one('data_files/aadr.us.txt')
d2 <- load_one('data_files/aaxj.us.txt')
# extract specified data column
d1_short <- d1[,c("Date","Open")]
d2_short <- d2[,c("Date","Open")]
# rename column based on symbol
names(d1_short)[2]<-d1[1, "Symbol"]
names(d2_short)[2]<- d2[1, "Symbol"]
# combine dataframes by date
d12 <- merge(d1_short, d2_short, by="Date", all=T)
# reshape dataframe for plotting
d12_plot <- melt(d12, id.vars="Date")
# generate plot
ggplot(d12_plot, aes(Date, value, col=variable)) +
  geom_line() +
  stat_smooth() +
```

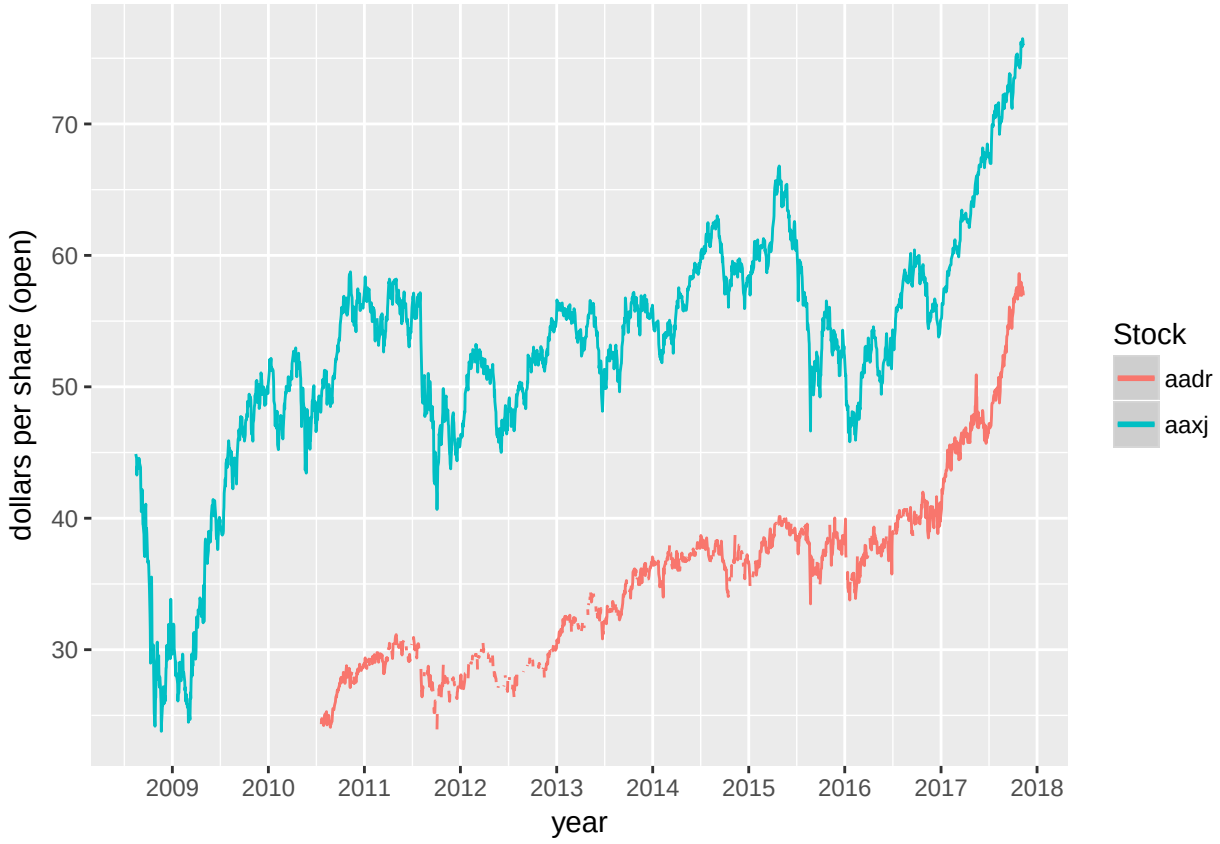
```
  ylab("dollars per share (open)") +
  guides(color=guide_legend("Stock")) +
  scale_x_date(name = 'year', date_breaks = '1 year',
               date_labels = '%Y')
```

```
## `geom_smooth()` using method = 'gam'
```

```
## Warning: Removed 762 rows containing non-finite values (stat_smooth).
```

```
## Warning: Computation failed in `stat_smooth()`:
## could not find function "s"
```

```
## Warning: Removed 483 rows containing missing values (geom_path).
```



Load a set of stock files.

The data are expected to be contained in CSV files (.txt). All files are combined into one dataframe of the chosen column. Only one data column is currently supported. (TODO extend) This function also extracts the stock symbol from the filename and includes it as a new column. References: - Function adapted from https://tinyurl.com/y8kypgpo - Map line from https://tinyurl.com/yc2htrp7 @param filepath name of location of stock files @param datacol name of the column to extract. This should be one of the following: “Open” “High” “Low” “Close” “Volume” “OpenInt” @return One dataframe with all files’ information @examples load_many(‘data_files’,“Open”) @export

```
load_many <- function(filepath,datacol){
  filenames <- list.files(filepath, pattern = "\\.txt$", full.names=TRUE)
  symlist   <- lapply(filenames, function(x){str_extract(basename(x), "(.*?)(?=\\.)")})
  datalist <- lapply(filenames, function(x){read.csv(x)[,c("Date",datacol)]})
  datalist <- do.call(rbind, unname(Map(cbind, Symbol = symlist, datalist)))
  # convert date column to date objects
```

```
  datalist$Date <- as.Date(datalist$Date)
  return(datalist)
}


### move towards slightly larger set
dmany <- load_many('data_files',"Open")
str(dmany)
```

```
## 'data.frame':    14004 obs. of  3 variables:
##  $ Symbol: Factor w/ 6 levels "aadr","aaxj",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Date  : Date, format: "2010-07-21" "2010-07-22" ...
##  $ Open  : num  24.3 24.6 24.8 24.6 24.5 ...
```

```
str(dmany)
```

```
## 'data.frame':    14004 obs. of  3 variables:
##  $ Symbol: Factor w/ 6 levels "aadr","aaxj",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Date  : Date, format: "2010-07-21" "2010-07-22" ...
##  $ Open  : num  24.3 24.6 24.8 24.6 24.5 ...
```

```
head(dmany)
```

```
##   Symbol       Date   Open
## 1   aadr 2010-07-21 24.333
## 2   aadr 2010-07-22 24.644
## 3   aadr 2010-07-23 24.759
## 4   aadr 2010-07-26 24.624
## 5   aadr 2010-07-27 24.477
## 6   aadr 2010-07-28 24.477
```

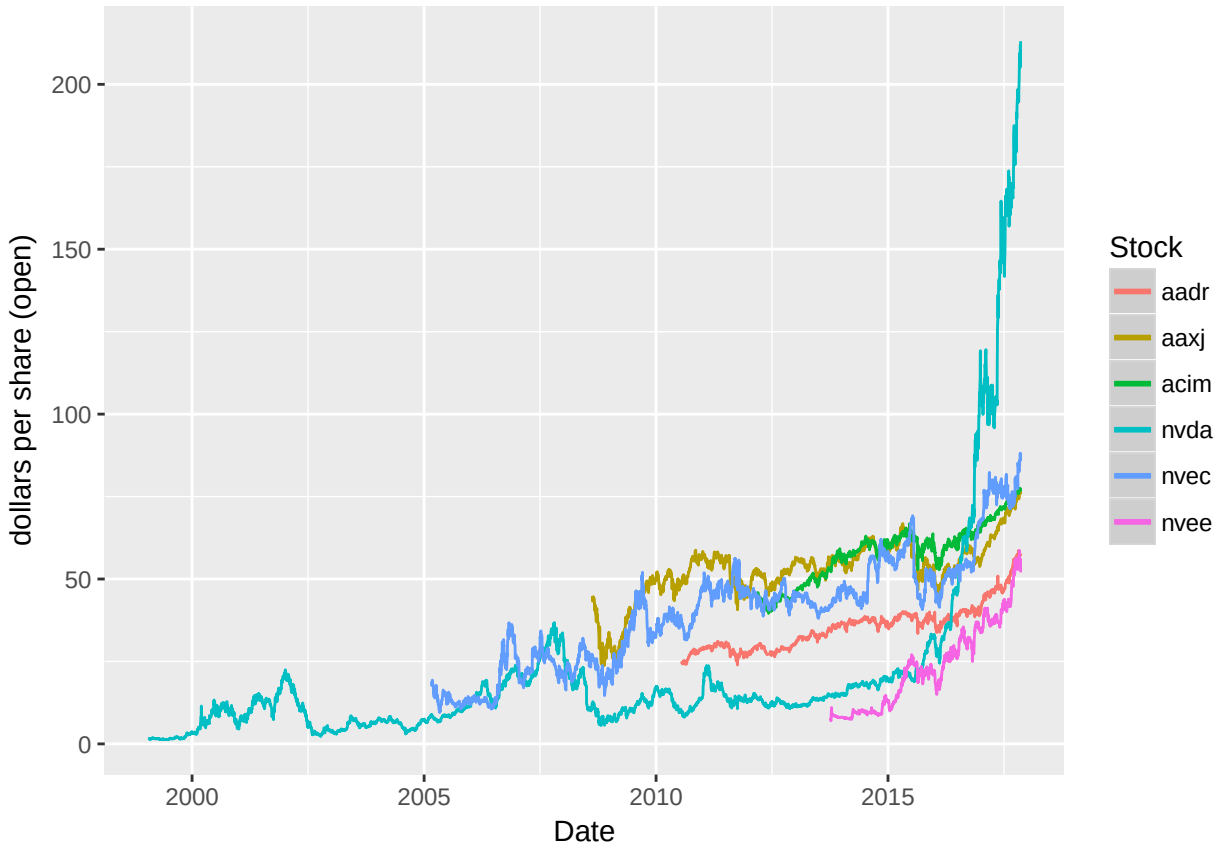
```
tail(dmany)
```

```
##       Symbol       Date  Open
## 13999   nvee 2017-11-03 55.50
## 14000   nvee 2017-11-06 55.60
## 14001   nvee 2017-11-07 56.15
## 14002   nvee 2017-11-08 55.00
## 14003   nvee 2017-11-09 55.70
## 14004   nvee 2017-11-10 52.00
```

```
# generate plot
ggplot(dmany, aes(Date, Open, col=Symbol)) +
  geom_line() +
  stat_smooth() +
  ylab("dollars per share (open)") +
  guides(color=guide_legend("Stock"))
```

```
## `geom_smooth()` using method = 'gam'
```

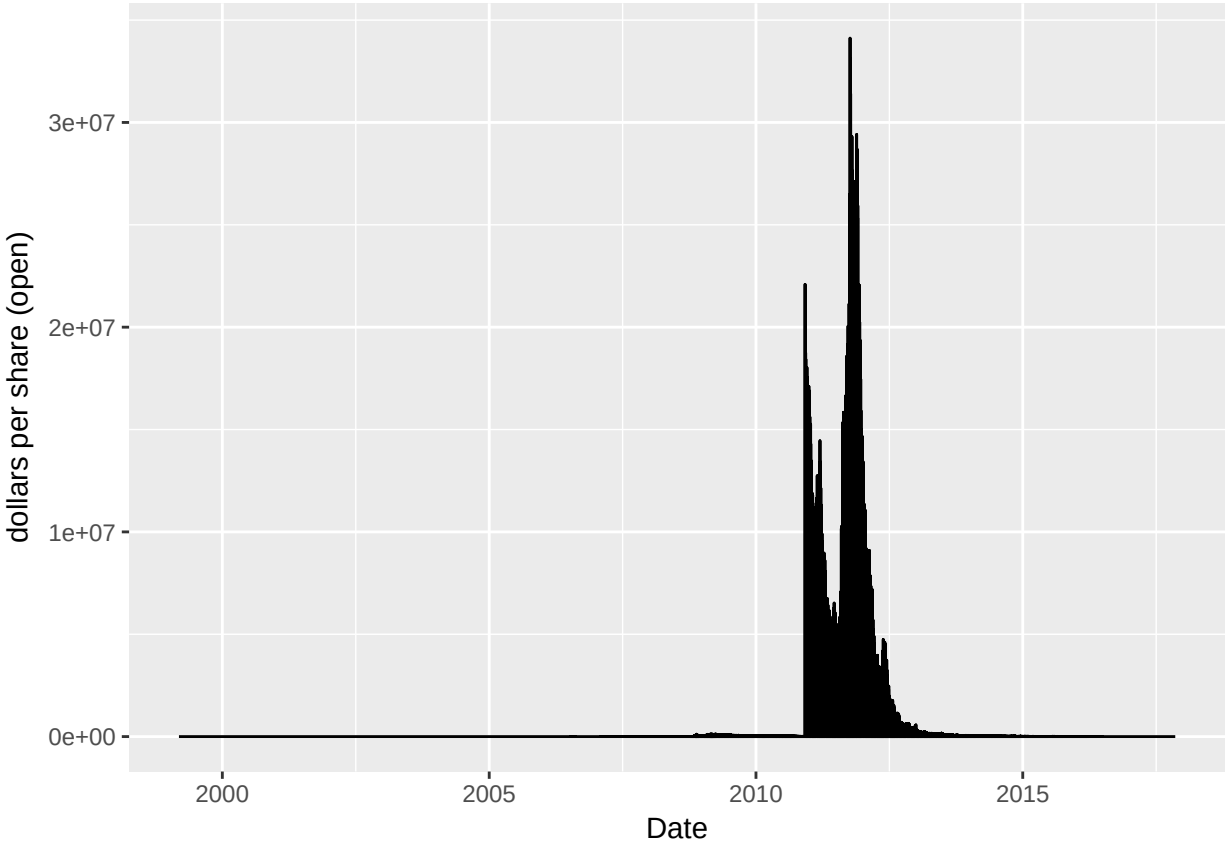
```
## Warning: Computation failed in `stat_smooth()`:
## could not find function "s"
```




```
### move towards whole set of ETFs [intensive!]
dmany2 <- load_many('ETFs',"Open")
# generate plot
ggplot(dmany2, aes(Date, Open)) +
  geom_line() +
  stat_smooth() +
  ylab("dollars per share (open)") +
  guides(color=guide_legend("Stock"))
```

```
## `geom_smooth()` using method = 'gam'
```

```
## Warning: Computation failed in `stat_smooth()`:
## could not find function "s"
```


3e+07
2e+07
1e+07
0e+00
dollars per share (open)
2000
2005
2010
2015
Date