Importing text files

IMPORTING AND MANAGING FINANCIAL DATA IN R

Joshua Ulrich
Instructor

datacamp

getSymbols() with CSV files

- Well-formatted
  - One instrument per file
  - Columns: date, open, high, low, close, volume, adjusted close
- Files named `"[symbol].csv"`
- Can use `dir` argument to specify directory

getSymbols() with CSV files

- AMZN.csv

```
"Date","AMZN.Open","AMZN.High","AMZN.Low","AMZN.Close","AMZN.Volume","AMZN.Adjusted"
2002-01-02,11.13,11.01,10.46,10.87,6674703,10.87
2002-01-03,11.26,12.25,10.76,11.99,11441553,11.99
2002-01-04,12.46,12.62,11.71,12.1,12619402,12.1
```

```
getSymbols("AMZN", src = "csv")
```

```
"AMZN"
```

```
head(AMZN, 3)
```

```
           AMZN.Open AMZN.High AMZN.Low AMZN.Close AMZN.Volume AMZN.Adjusted
2002-01-02     11.13     11.01    10.46      10.87     6674703         10.87
2002-01-03     11.26     12.25    10.76      11.99    11441553         11.99
2002-01-04     12.46     12.62    11.71      12.10    12619402         12.10
```

read.zoo()

- AMZN.csv

```
"Date","AMZN.Open","AMZN.High","AMZN.Low","AMZN.Close","AMZN.Volume","AMZN.Adjusted"
2002-01-02,11.13,11.01,10.46,10.87,6674703,10.87
2002-01-03,11.26,12.25,10.76,11.99,11441553,11.99
2002-01-04,12.46,12.62,11.71,12.1,12619402,12.1
```

```
amzn_zoo <- read.zoo("AMZN.csv", sep = ",", header = TRUE)
amzn_xts <- as.xts(amzn_zoo)
head(amzn_xts, n = 3)
```

```
           AMZN.Open AMZN.High AMZN.Low AMZN.Close AMZN.Volume AMZN.Adjusted
2002-01-02     11.13     11.01    10.46      10.87     6674703         10.87
2002-01-03     11.26     12.25    10.76      11.99    11441553         11.99
2002-01-04     12.46     12.62    11.71      12.10    12619402         12.10
```

Date and time in separate columns

- FOO.csv

```
"Date","Time","Open","High","Low","Close"
2016-11-08,09:05:00,80.9,81,80.87,81
2016-11-08,09:10:00,80.92,80.93,80.89,80.89
2016-11-08,09:15:00,80.93,80.94,80.92,80.93
```

```
foo_zoo <- read.zoo("FOO.csv", sep = ",", header = TRUE,
                    index.column = c("Date", "Time"))
head(foo_zoo,  n = 3)
```

```
                     Open  High   Low Close
2016-11-08 09:05:00 80.90 81.00 80.87 81.00
2016-11-08 09:10:00 80.92 80.93 80.89 80.89
2016-11-08 09:15:00 80.93 80.94 80.92 80.93
```

File contains multiple instruments

- BAR.csv

```
Date,Symbol,Type,Price
2016-01-01 10:43:01,A,Bid,58.23
2016-01-01 10:43:01,A,Ask,58.24
2016-01-01 10:43:01,B,Bid,28.96
2016-01-01 10:43:01,B,Ask,28.98
```

```
bar_zoo <- read.zoo("BAR.csv",
                    split = c("Symbol", "Type"),
                    sep = ",", header = TRUE)
bar_zoo
```

```
                    A.Ask B.Ask A.Bid B.Bid
2016-01-01 10:43:01 58.24 28.98 58.23 28.96
2016-01-01 10:43:02 58.25 28.99 58.24 28.97
```

Let's practice!

IMPORTING AND MANAGING FINANCIAL DATA IN R

Checking for weirdness

IMPORTING AND MANAGING FINANCIAL DATA IN R

Joshua Ulrich
Instructor

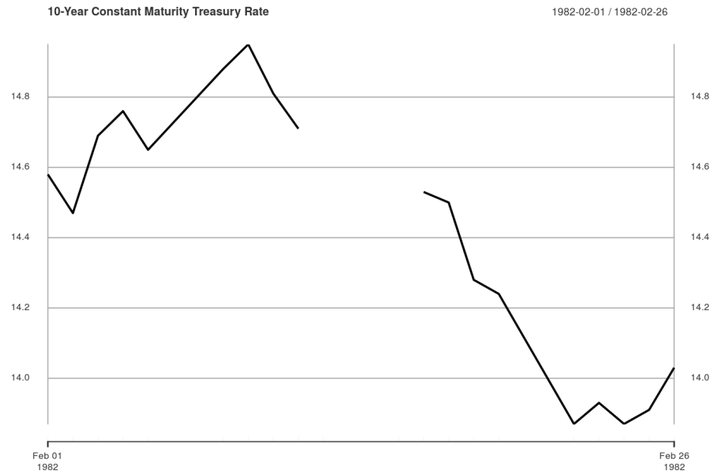
datacamp

Visualize Data

```
getSymbols("DGS10", src = "FRED")
```

```
"DGS10"
```

```
treasury_10 <- DGS10["1982-02"]
plot(treasury_10, main = "10-Year Constant Maturity Treasury Rate")
```



Handle missing values

```
# Fill NA using last observation carried forward
locf <- na.locf(treasury_10)
```

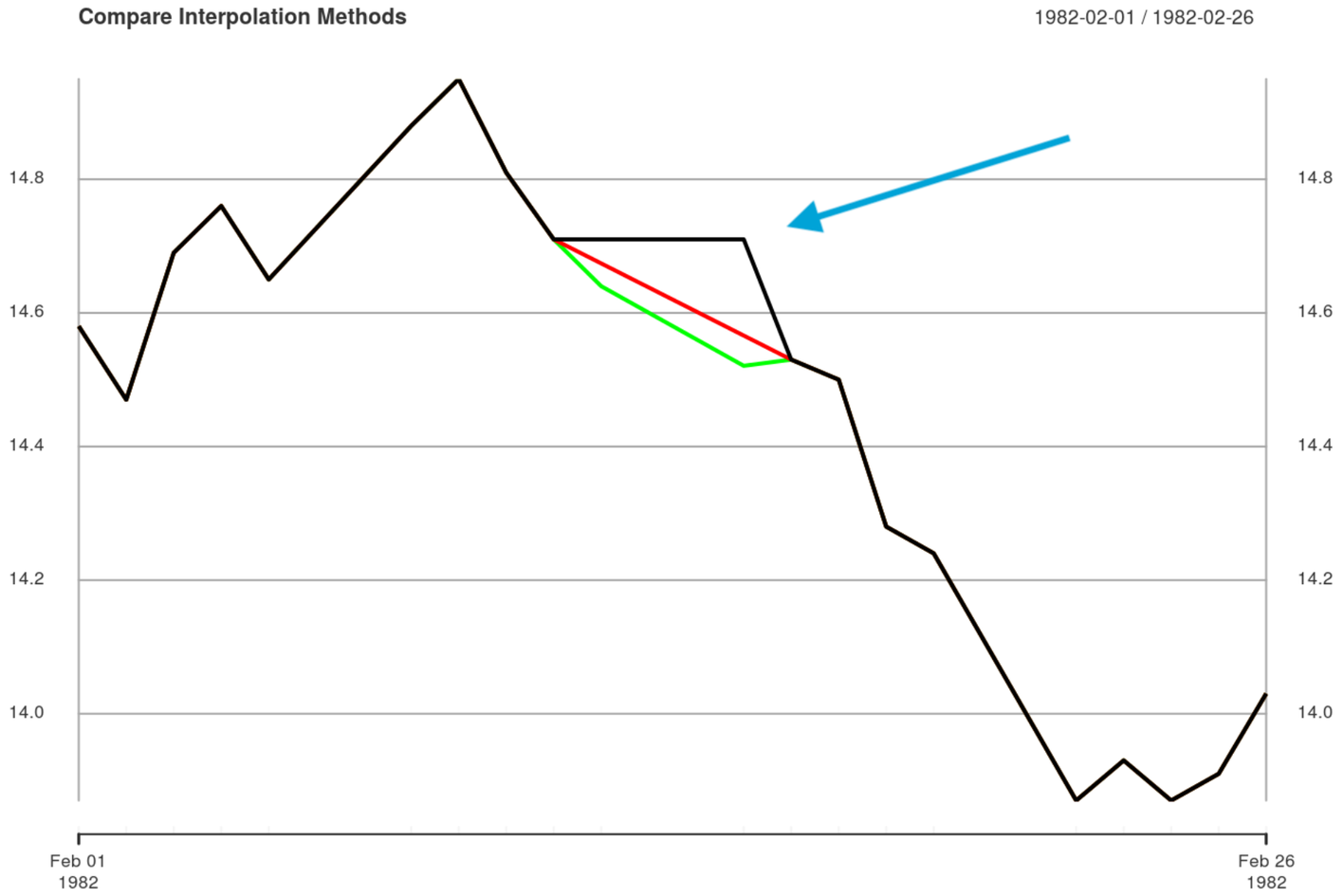
```
# Fill NA using linear interpolation
approx <- na.approx(treasury_10)
```

```
# Fill NA using spline interpolation
spline <- na.spline(treasury_10)
```

```
# Merge into one object
na_filled <- merge(locf, approx, spline)
# Plot combined object
plot(na_filled, col = c("black", "red", "green"),
     main = "Compare Interpolation Methods")
```

Handle missing values

Handle missing values

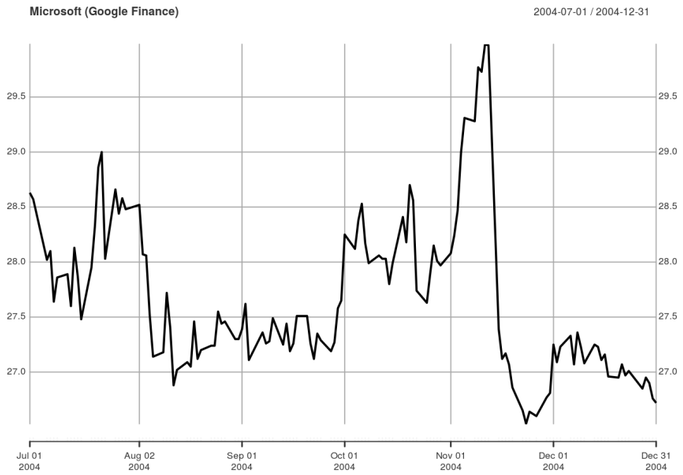


Visualize data

```
getSymbols("MSFT", from = "2004-07-01", to = "2004-12-31", src = "google")
```

```
"MSFT"
```

```
plot(Cl(MSFT), main = "Microsoft (Google Finance)")
```

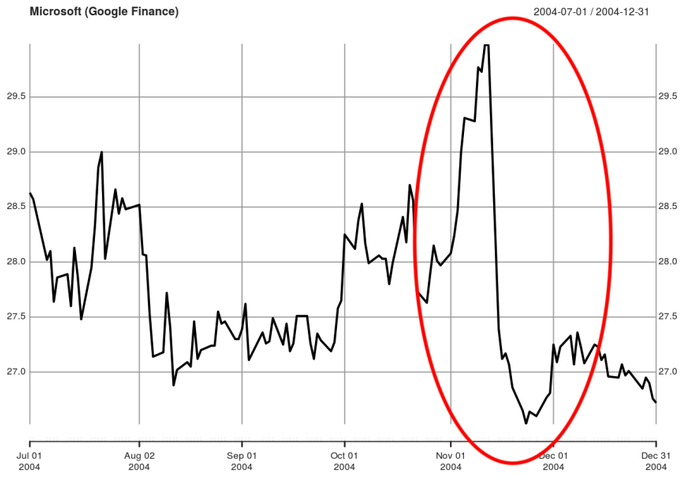


Visualize data

```
getSymbols("MSFT", from = "2004-07-01", to = "2004-12-31", src = "google")
```

```
"MSFT"
```

```
plot(Cl(MSFT), main = "Microsoft (Google Finance)")
```

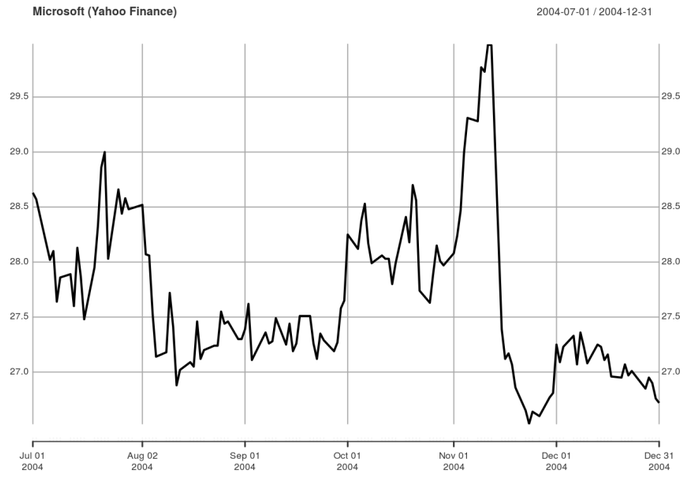


Cross-reference sources

```
getSymbols("MSFT", from = "2004-07-01", to = "2004-12-31")
```

```
"MSFT"
```

```
plot(Cl(MSFT), main = "Microsoft (Yahoo Finance)")
```

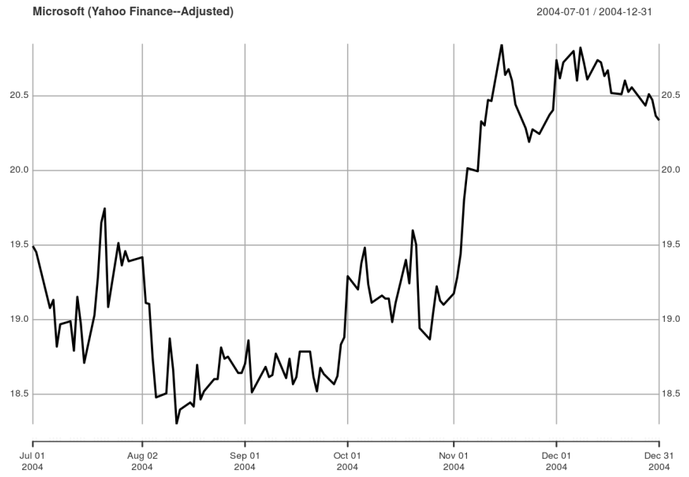


Cross-reference sources

```
getSymbols("MSFT", from = "2004-07-01", to = "2004-12-31")
```

```
"MSFT"
```

```
plot(Ad(MSFT), main = "Microsoft (Yahoo Finance-Adjusted)")
```

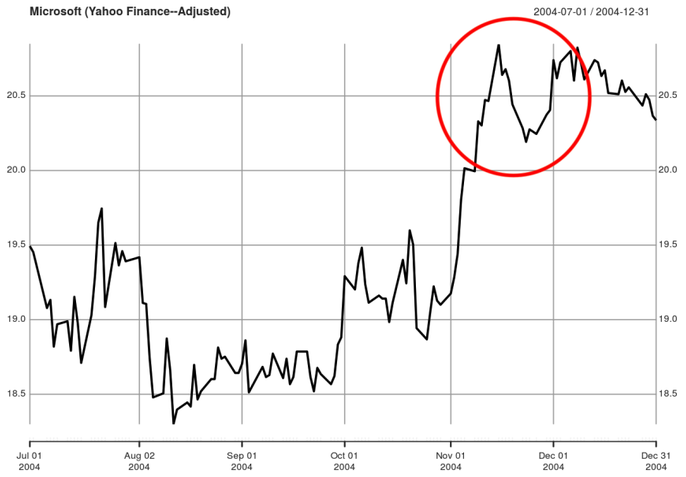


Cross-reference sources

```
getSymbols("MSFT", from = "2004-07-01", to = "2004-12-31")
```

```
"MSFT"
```

```
plot(Ad(MSFT), main = "Microsoft (Yahoo Finance—Adjusted)")
```



Stock split example

- MSFT stock splits 2-for-1

	Pre-split	Post-split
Shares	100	200
Price	$50	$25
Value	$5,000	$5,000

Stock dividend example

- MSFT issues a $3 per share dividend

	Pre-dividend	Post-dividend
Cash	$0	$300
Shares	100	100
Price	$50	$47
Value	$5,000	$5,000

Data source differences

- Yahoo Finance:
  - Raw OHLC prices
  - Split- and dividend-adjusted close
- Google Finance:
  - Split-adjusted OHLC prices

Let's practice!

IMPORTING AND MANAGING FINANCIAL DATA IN R

Adjusting for corporate actions

IMPORTING AND MANAGING FINANCIAL DATA IN R

Joshua Ulrich
Instructor

datacamp

Adjust for stock splits and dividends (1)

```
getSymbols("MSFT", from = "2004-07-01", to = "2004-12-31")
```

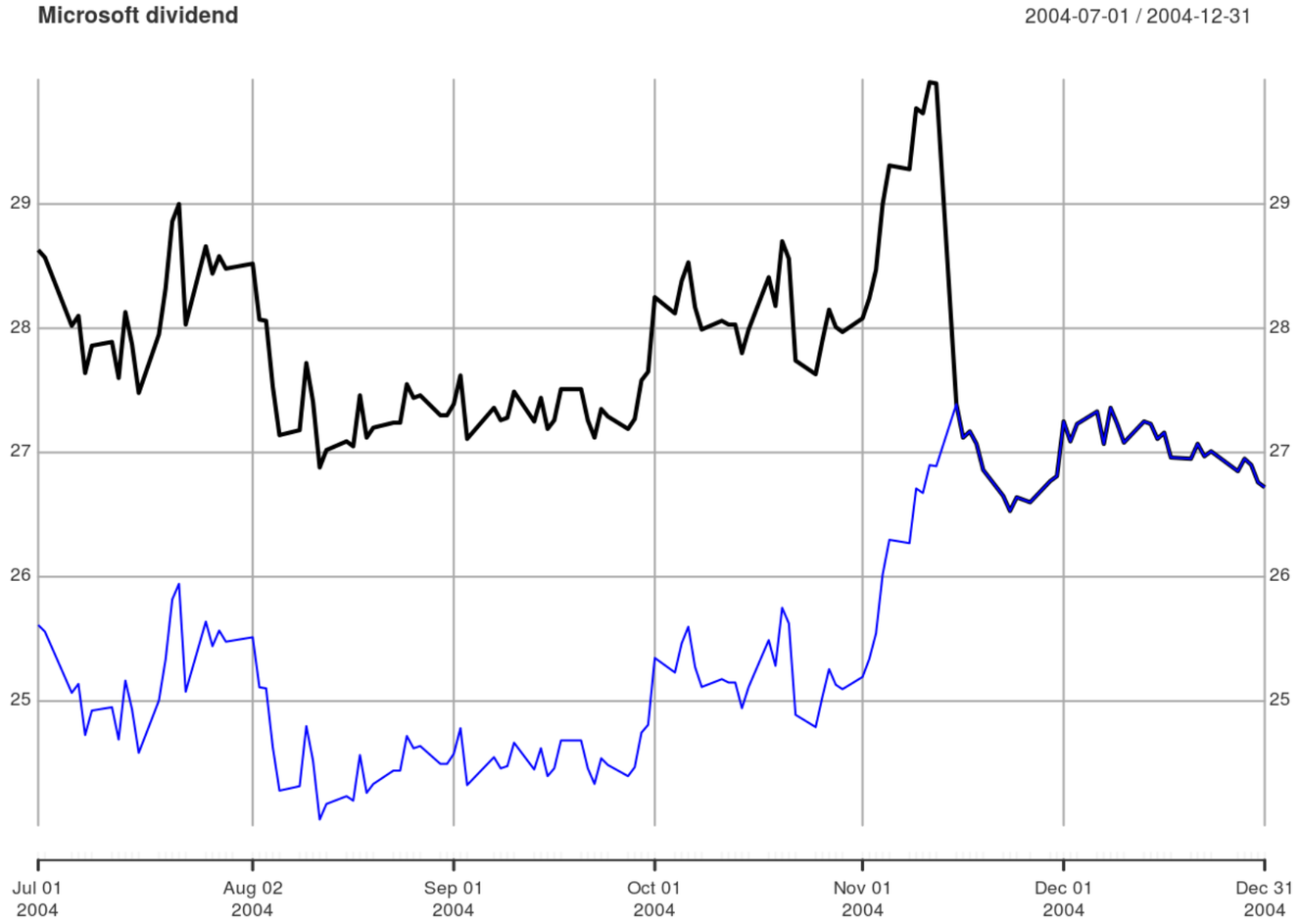
```
"MSFT"
```

```
# Adjust data for splits and dividends
msft_adjusted <- adjustOHLC(MSFT)
```

```
# Object name is not ticker symbol
my_data <- MSFT

# Use symbol.name argument
my_data_adjusted <- adjustOHLC(my_data, symbol.name = "MSFT")
```

Adjust for stock splits and dividends (2)



```r
# Download split data from Yahoo Finance
splits <- getSplits("GE")
head(splits, n = 4)
```

```
           GE.spl
1971-06-08    0.5
1983-06-02    0.5
1987-05-26    0.5
1994-05-16    0.5
```

```r
# Download split-adjusted dividend data from Yahoo Finance
dividends <- getDividends("GE")
head(dividends, n = 4)
```

```
            GE.div?1970-03-03 0.00677
1970-06-11 0.00677
1970-09-21 0.00677
1970-12-07 0.00677
```

Download unadjusted dividends

```
# Download unadjusted dividend data from Yahoo Finance
dividends_raw <- getDividends("GE", split.adjust = FALSE)

# Compare adjusted and unadjusted dividends
head(merge(dividends, dividends_raw))
```

```
            GE.div GE.div.1
1970-03-03 0.00677  0.64992
1970-06-11 0.00677  0.64992
1970-09-21 0.00677  0.64992
1970-12-07 0.00677  0.64992
1971-03-03 0.00677  0.64992
1971-06-17 0.00729  0.34992
```

adjRatios()

- Back-adjust any series for splits, dividends, or both
- Has 3 arguments:
  - `splits`
  - `dividends`
  - `close`
- Returns `xts` object with 2 columns: `Split` and Div

Adjust univariate series for splits and dividends

```
getSymbols("GE", from = "2000-01-01")
```

```
"GE"
```

```
close <- Cl(GE)
splits <- getSplits("GE")
dividends_raw <- getDividends("GE", split.adjust = FALSE)
```

```
# Pass splits, unadjusted dividends, and unadjusted close
ratios <- adjRatios(splits = splits,
                    dividends = dividends_raw,
                    close = close)
```

Adjust univariate series for splits and dividends

```
# Multiply unadjusted close by split and dividend ratios
close_adjusted <- close * ratios[, "Split"] * ratios[, "Div"]

head(merge(close, close_adjusted, Ad(GE)), n = 4)
```

```
           GE.Close GE.Close.1 GE.Adjusted
2000-01-03 150.0000   29.50422    29.44630
2000-01-04 144.0000   28.32405    28.26845
2000-01-05 143.7500   28.27488    28.21937
2000-01-06 145.6718   28.65289    28.59664
```

Let's practice!

IMPORTING AND MANAGING FINANCIAL DATA IN R

Congratulations!

IMPORTING AND MANAGING FINANCIAL DATA IN R

Joshua Ulrich
Instructor

Congratulations!

IMPORTING AND MANAGING FINANCIAL DATA IN R