Why use apply?

INTERMEDIATE R FOR FINANCE

Lore Dirick

Manager of Data Science Curriculum at Flatiron School

datacamp

Meet the apply family

Function	Description
apply	Apply functions over array margins
lapply	Apply a function over a list or vector
eapply	Apply a function over values in an environment
mapply	Apply a function to multiple lists or vector arguments
rapply	Recursively apply a function to a list
tapply	Apply a function over a ragged array
sapply	Simplify the result from lapply
vapply	Strictly simplify the result from lapply

Meet the apply family

Function	Description
apply	Apply functions over array margins
lapply	**Apply a function over a list or vector**
eapply	Apply a function over values in an environment
mapply	Apply a function to multiple lists or vector arguments
rapply	Recursively apply a function to a list
tapply	Apply a function over a ragged array
sapply	**Simplify the result from lapply**
vapply	**Strictly simplify the result from lapply**

Meet the apply family

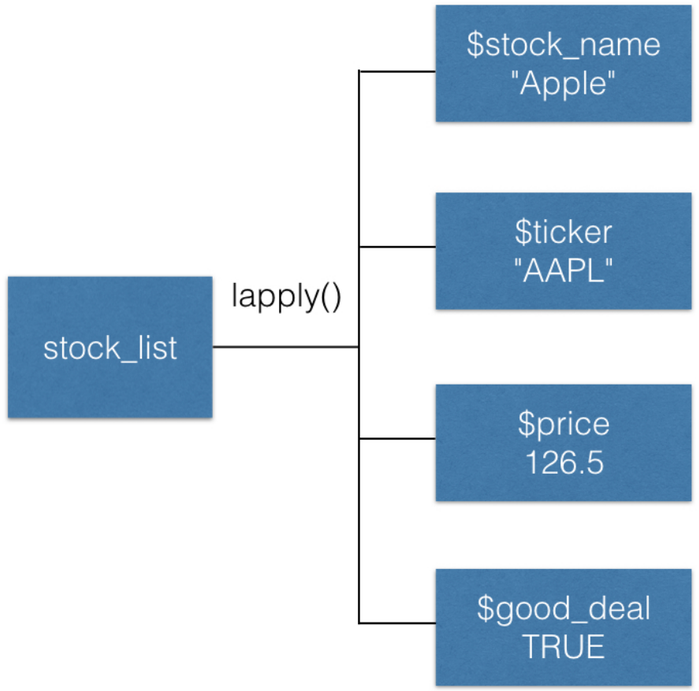
Function	Description
apply	Apply functions over array margins
lapply	Apply a function over a list or vector
eapply	Apply a function over values in an environment
mapply	Apply a function to multiple lists or vector arguments
rapply	Recursively apply a function to a list
tapply	Apply a function over a ragged array
sapply	Simplify the result from lapply
vapply	Strictly simplify the result from lapply

lapply()

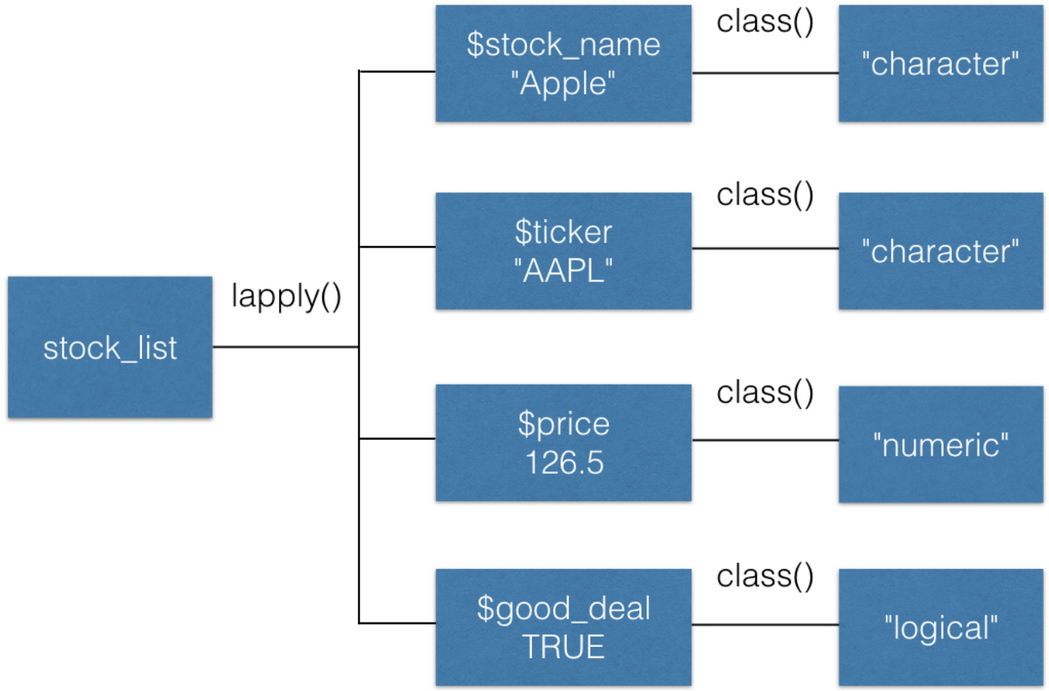
```
stock_list <- list(stock_name = "Apple", ticker = "AAPL",
                   price = 126.5, good_deal = TRUE)
lapply(stock_list, FUN = class)
```

```
$stock_name
"character"

$ticker
"character"

$price
"numeric"

$good_deal
"logical"
```

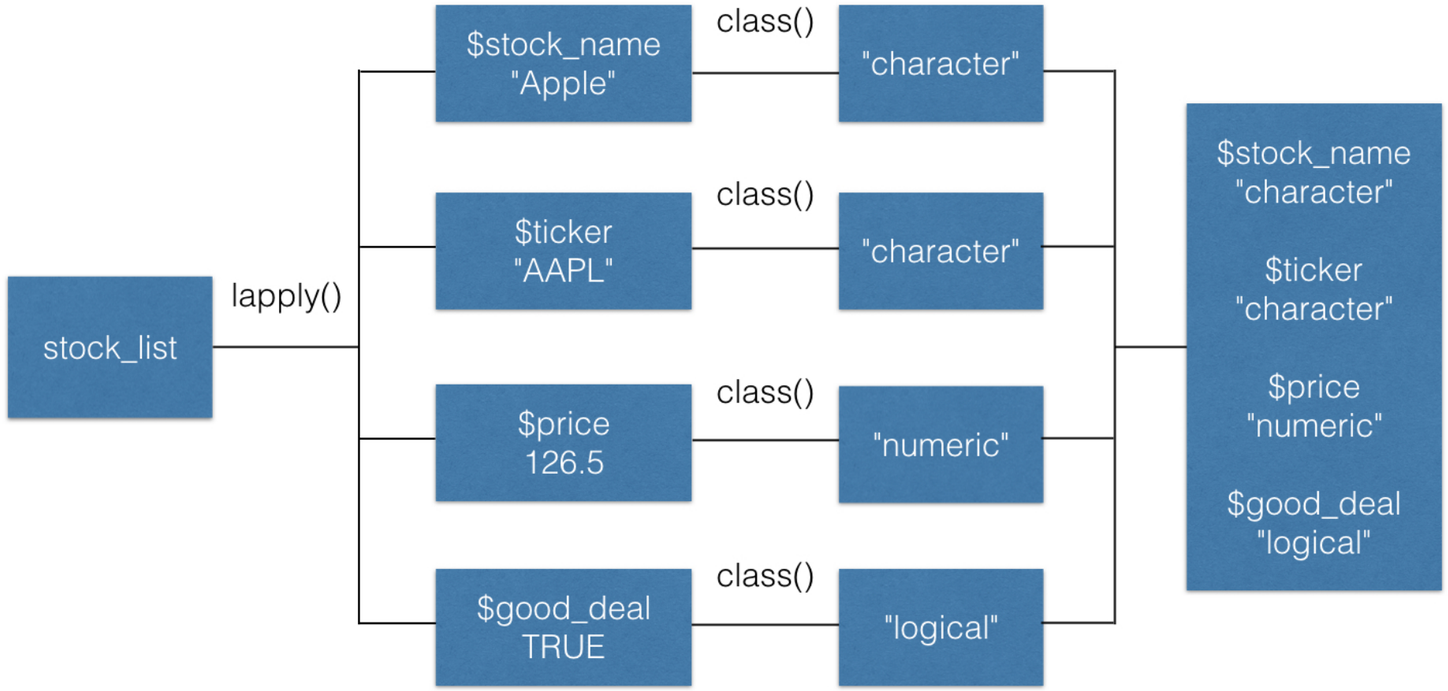
Break it down



Break it down



Break it down



Sharpe ratio

$$sharpe = \frac{mean(r) - r_f}{sd(r)}$$

- Normalize returns by risk
- Compare returns among stocks
- Higher sharpe ratio = More return / unit risk

Let's practice!

INTERMEDIATE R FOR FINANCE

sapply() - simplify it!

INTERMEDIATE R FOR FINANCE

Lore Dirick

Manager of Data Science Curriculum at Flatiron School

datacamp

sapply()

```
stock_list <- list(stock_name = "Apple", ticker = "AAPL",
                   price = 126.5, good_deal = TRUE)
sapply(stock_list, FUN = class)
```

```
 stock_name       ticker        price    good_deal
"character" "character"    "numeric"    "logical"
```

Apply a custom summary function

```
simple_summary <- function(x) {
  c(mean = mean(x), sd = sd(x))
}
head(stock_return, 3)
```

```
         apple          ibm          micr
1  0.003744634  0.001251408  0.0008445946
2 -0.007188353 -0.001124859  0.0163713080
3  0.007698653  0.003190691 -0.0044835603
```

```
sapply(stock_return, FUN = simple_summary)
```

```
           apple         ibm        micr
mean 0.002838389 0.001926806 0.002472939
sd   0.007157457 0.008130703 0.009943938
```

Let's practice!

INTERMEDIATE R FOR FINANCE

vapply() - specify your output!

INTERMEDIATE R FOR FINANCE

Dan Becker

Manager of Data Science Curriculum at Flatiron School

vapply()

```
args(vapply)
```

```
function (X, FUN, FUN.VALUE, ..., USE.NAMES = TRUE)
NULL
```

```
vapply(stock_list, FUN = class, FUN.VALUE = character(1))
```

```
 stock_name      ticker       price   good_deal
"character" "character"   "numeric"   "logical"
```

Anonymous functions

```
vapply(stock_return,
       FUN = function(x) {c(mean = mean(x), sd = sd(x))},
       FUN.VALUE = numeric(2))
```

```
            apple          ibm         micr
mean 0.002838389 0.001926806 0.002472939
sd   0.007157457 0.008130703 0.009943938
```

Let's practice!

INTERMEDIATE R FOR FINANCE

Congratulations

INTERMEDIATE R FOR FINANCE

Lore Dirick

Manager of Data Science Curriculum at Flatiron School

Popular R packages in Finance

- `quantmod`
- `xts` / `zoo`
- `forecast`
- `quantstrat`
- `Quandl`
- `PerformanceAnalytics`
- Empirical Finance Task View on CRAN

Let's practice!

INTERMEDIATE R FOR FINANCE