What are functions?

INTERMEDIATE R FOR FINANCE

Lore Dirick

Manager of Data Science Curriculum at Flatiron School

Examples of functions

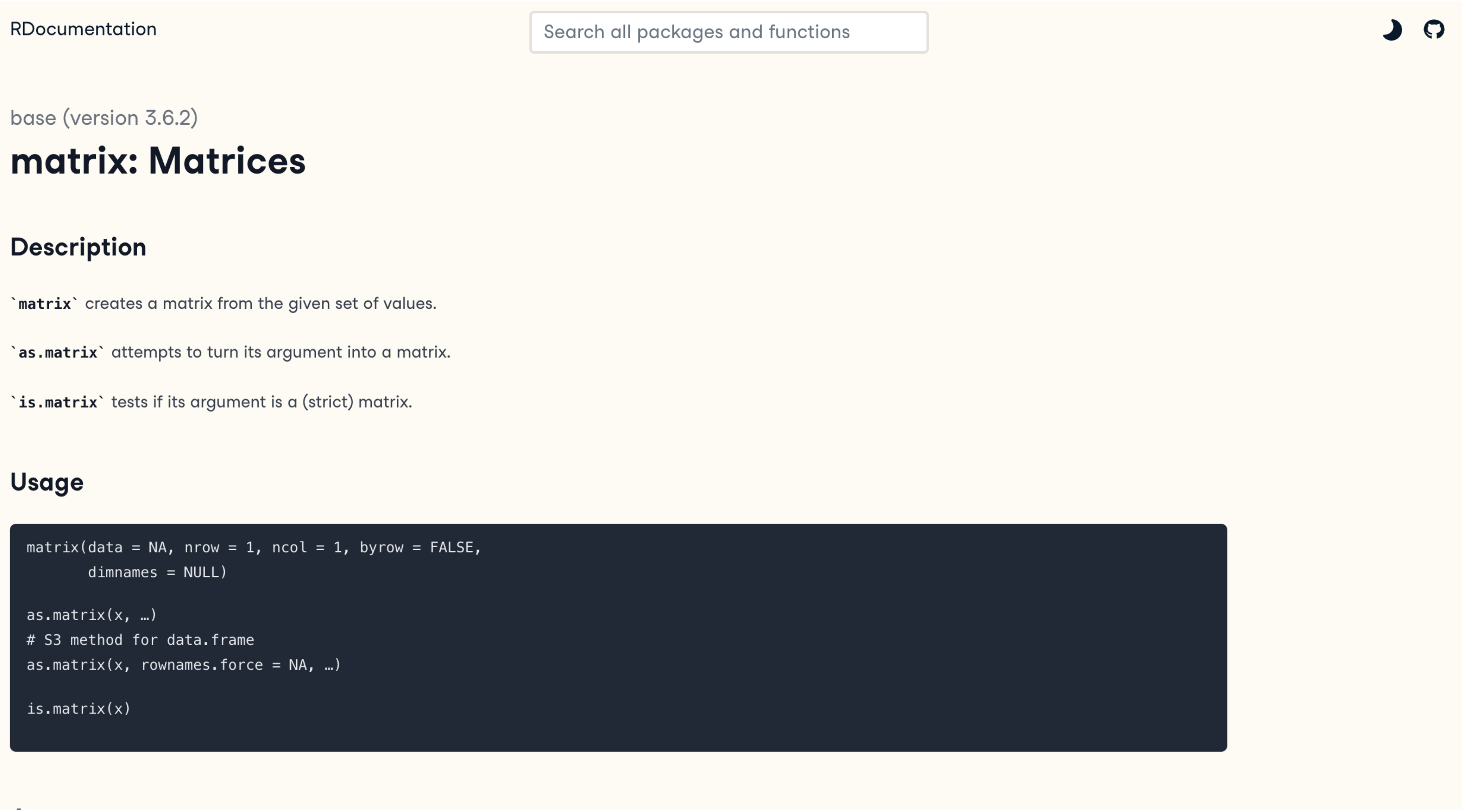
- `mean()`
- `plot()`
- `ncol()`

Elements of a function

- Arguments
  - Input / data
  - Options
- Body
  - Code execution
- Return
  - Stable and predicable output

Function documentation

```
?matrix
```



base (version 3.6.2)

matrix: Matrices

Description

`` `matrix` `` creates a matrix from the given set of values.

`` `as.matrix` `` attempts to turn its argument into a matrix.

`` `is.matrix` `` tests if its argument is a (strict) matrix.

Usage

```
matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE,
       dimnames = NULL)

as.matrix(x, …)
# S3 method for data.frame
as.matrix(x, rownames.force = NA, …)

is.matrix(x)
```

Function arguments

- Required
  - Error thrown without it
  - Normally data / object
- Optional
  - Default values are set
  - Normally sets extra options

Function arguments example

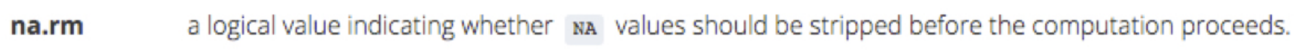
```
returns <- c(.023, .044, .034, NA)
mean()
```

```
Error in mean.default() : argument "x" is missing, with no default
```

```
mean(returns)
```

```
NA
```

```
?mean
```



na.rm a logical value indicating whether NA values should be stripped before the computation proceeds.

```
mean(returns, na.rm = TRUE)
```

```
0.03366667
```

Let's practice!

INTERMEDIATE R FOR FINANCE

Writing functions

INTERMEDIATE R FOR FINANCE

Lore Dirick

Manager of Data Science Curriculum at Flatiron School

Function structure

```
func_name <- function(arguments) {
  body
}
```

Add one

```
add_one <- function(x) {
  x_plus_one <- x + 1
  return(x_plus_one)
}
add_one(7)
```

```
8
```

```
add_one <- function(x) {
  x + 1
}
add_one(7)
```

```
8
```

Using an optional argument

```
add <- function(x, value = 1) {
  x + value
}
add(7)
```

```
8
```

```
add(7, value = 3)
```

```
10
```

Calculating arithmetic returns

$$\frac{S_t - S_{t-1}}{S_{t-1}}$$

```
prices <- c(23.4, 23.8, 22.3)
# S_(t) - S(t-1) vector
diff(prices)
```

```
0.4 -1.5
```

```
# S_(t-1) vector
prices[-length(prices)]
```

```
23.4 23.8
```

```
# Arithmetic returns
diff(prices) / prices[-length(prices)]
```

```
0.01709402 -0.06302521
```

Calculating arithmetic returns

```r
prices <- c(23.4, 23.8, 22.3)
arith_returns <- function(x) {
  diff(x) / x[-length(x)]
}
arith_returns(prices)
```

```
0.01709402 -0.06302521
```

Let's practice!

INTERMEDIATE R FOR FINANCE

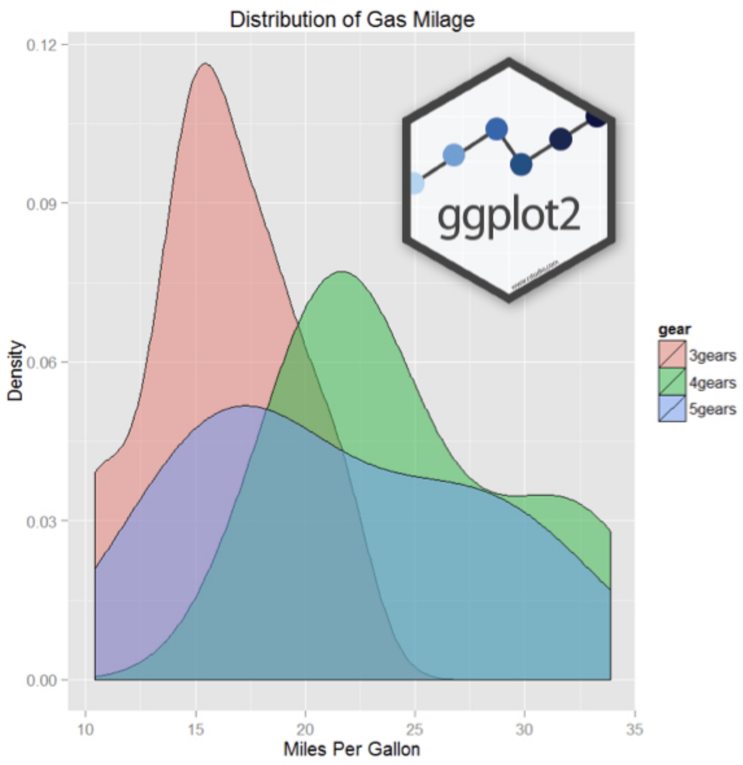
Packages

INTERMEDIATE R FOR FINANCE

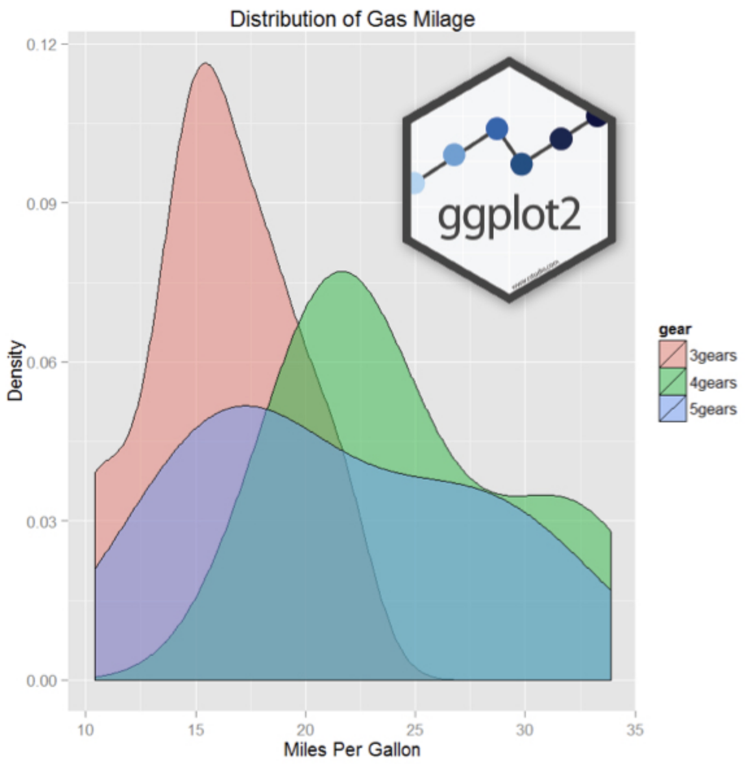
Lore Dirick

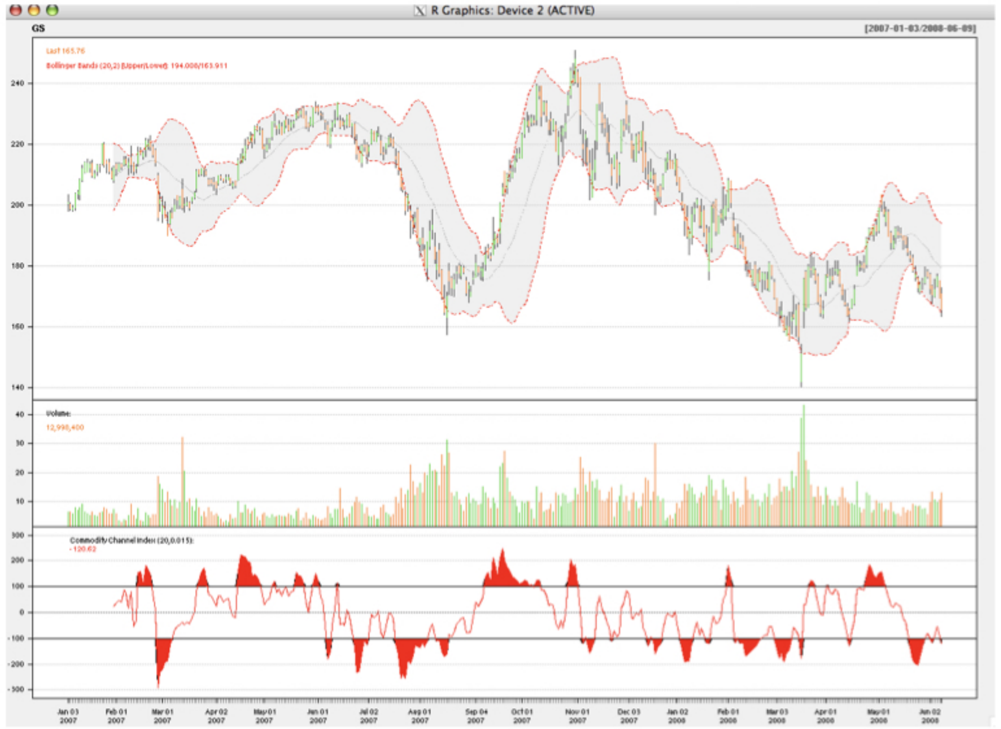
Manager of Data Science Curriculum at Flatiron School

Packages


Distribution of Gas Milage
0.12
0.09
0.06
0.03
0.00
Density
10
15
20
25
30
35
Miles Per Gallon
ggplot2
gear
3gears
4gears
5gears

Packages


Distribution of Gas Milage
Density
0.12
0.09
0.06
0.03
0.00
10
15
20
25
30
35
Miles Per Gallon
gear
3gears
4gears
5gears
ggplot2


R Graphics: Device 2 (ACTIVE)
GS
[2007-01-03/2008-06-09]
Last 165.76
Bollinger Bands (20,2) [Upper/Lower]: 194.006/163.911
240
220
200
180
160
140
Volume:
12,998,400
40
30
20
10
Commodity Channel Index (20,0.015):
-120.62
300
200
100
0
-100
-200
-300
Jan 03 2007
Feb 01 2007
Mar 01 2007
Apr 02 2007
May 01 2007
Jun 01 2007
Jul 02 2007
Aug 01 2007
Sep 04 2007
Oct 01 2007
Nov 01 2007
Dec 03 2007
Jan 02 2008
Feb 01 2008
Mar 03 2008
Apr 01 2008
May 01 2008
Jun 02 2008

CRAN

- Comprehensive R Archive Network
- More than 10000 packages

Installing packages

```
# Download from CRAN
install.packages("quantmod")
```

```
# Load into your current R session
library(quantmod)
```

quantmod functionality

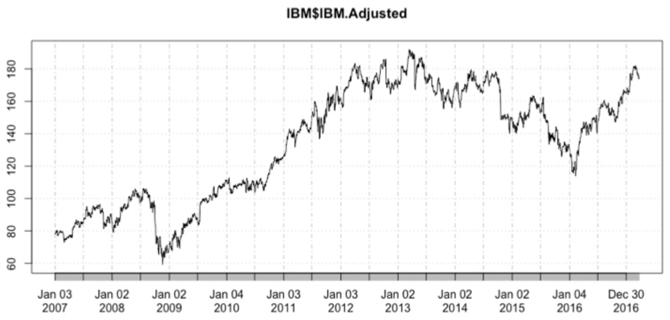
```r
library(quantmod)
getSymbols("IBM")
```

```
"IBM"
```

```r
head(IBM, n = 3)
```

```
           IBM.Open IBM.High IBM.Low IBM.Close IBM.Volume IBM.Adjusted
2007-01-03    97.18    98.40   96.26     97.27    9196800     77.73997
2007-01-04    97.25    98.79   96.88     98.31   10524500     78.57116
2007-01-05    97.60    97.95   96.91     97.42    7221300     77.85985
```

```r
plot(IBM$IBM.Adjusted)
```



Let's practice!

INTERMEDIATE R FOR FINANCE