INTERMEDIATE R FOR FINANCE

What are functions?

Lore Dirick
Instructor, DataCamp

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
```

RDocumentation | Search for packages, functions, etc | R package | Leaderboard | Sign in

matrix

From base v3.3.2
by R-core R-core@R-project.org

Matrices

`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.

Keywords algebra, array

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)
```

Arguments

data an optional data vector (including a list or `expression` vector). Non-atomic classed R objects are coerced by `as.vector` and all attributes discarded.

nrow the desired number of rows.

ncol the desired number of columns.

byrow logical. If `FALSE` (the default) the matrix is filled by columns, otherwise the matrix is filled by rows.

dimnames A `dimnames` attribute for the matrix: `NULL` or a `list` of length 2 giving the row and column names respectively. An empty list is treated as `NULL`, and a list of

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)
[1] NA


> ?mean
```

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

```
> mean(returns, na.rm = TRUE)
[1] 0.03366667
```

INTERMEDIATE R FOR FINANCE

Let’s practice!

INTERMEDIATE R FOR FINANCE

Writing functions

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)
[1] 8
```

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

> add_one(7)
[1] 8
```

Using an optional argument

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

> add(7)
[1] 8

> add(7, value = 3)
[1] 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)
[1]  0.4 -1.5

> # S_(t-1) vector
> prices[-length(prices)]
[1] 23.4 23.8

> # Arithmetic returns
> diff(prices) / prices[-length(prices)]
[1]  0.01709402 -0.06302521
```

Calculating arithmetic returns

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

```
> prices <- c(23.4, 23.8, 22.3)

> arith_returns <- function(x) {
  diff(x) / x[-length(x)]
}

> arith_returns(prices)
[1]  0.01709402 -0.06302521
```

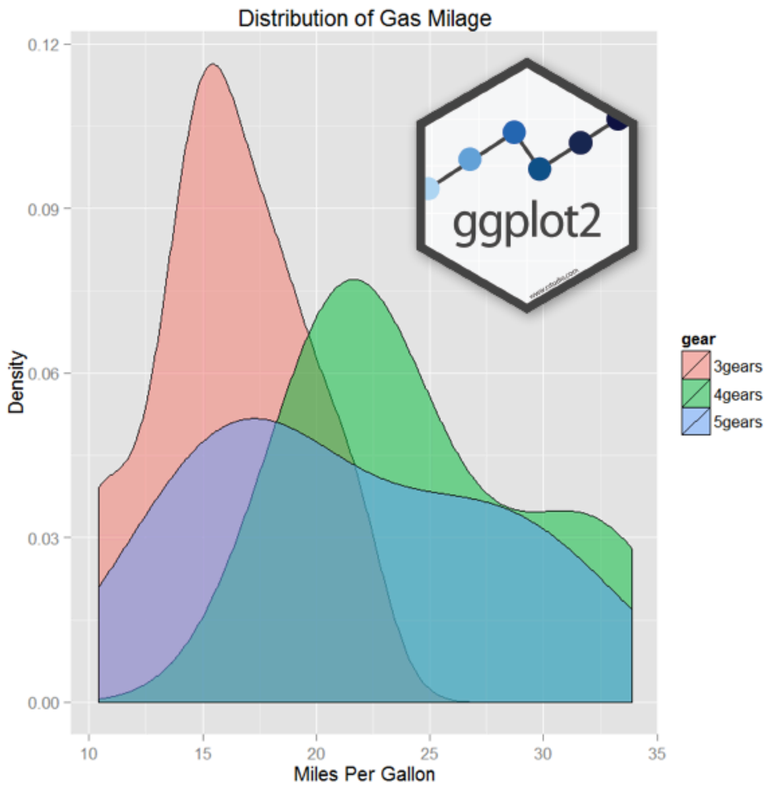
INTERMEDIATE R FOR FINANCE

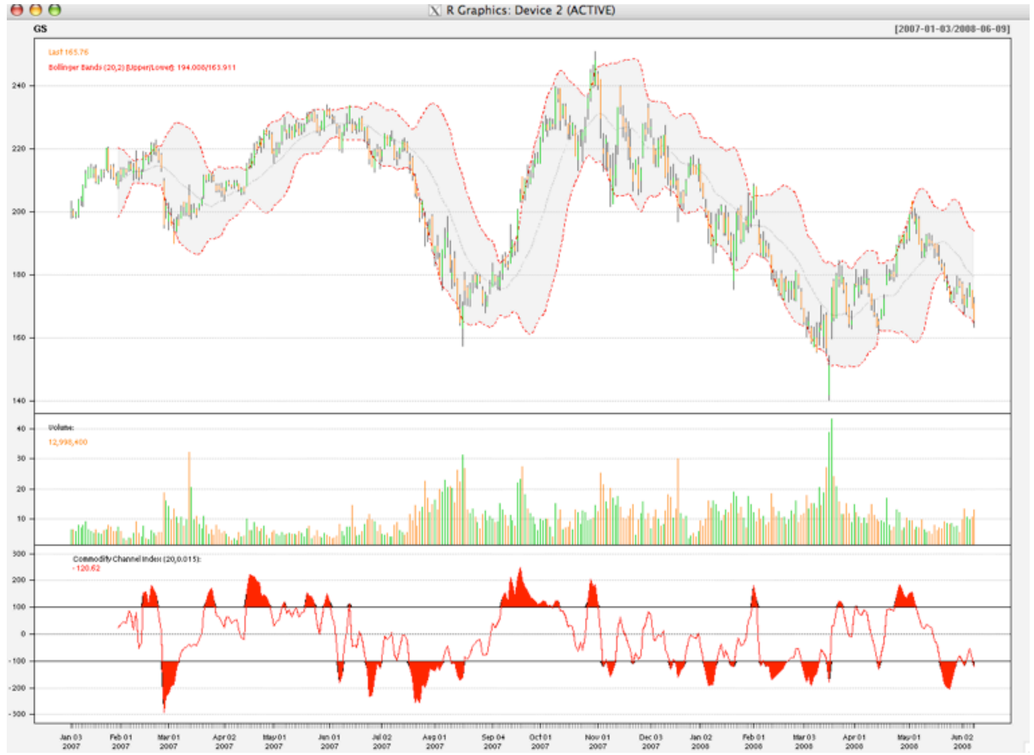
Let’s practice!

INTERMEDIATE R FOR FINANCE

Packages

Packages


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


R Graphics: Device 2 (ACTIVE)
GS
[2007-01-03/2008-06-09]

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)
```

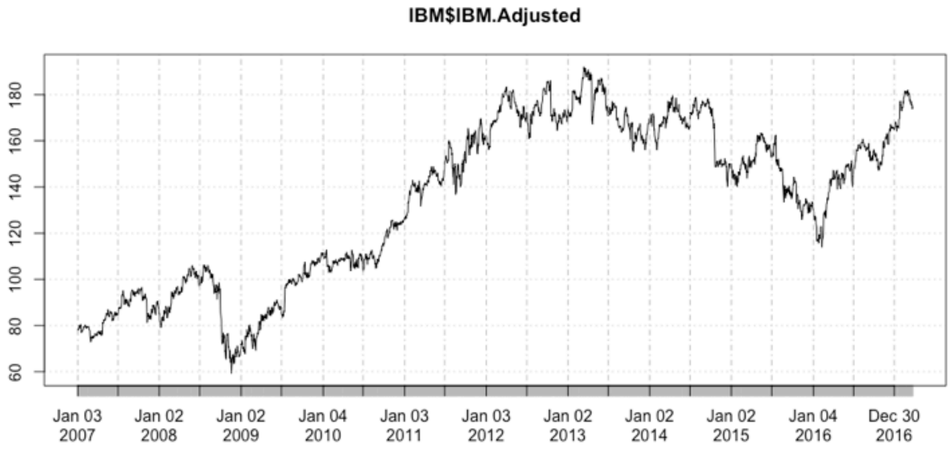
quantmod functionality

```
> library(quantmod)

> getSymbols("IBM")
[1] "IBM"

> 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

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



INTERMEDIATE R FOR FINANCE

Let’s practice!