A Beautiful Paradigm
Functional Programming in Finance

Brian Lee Yung Rowe
Director, Engineering and Product
trueSEF

brian@truesef.com
Proof by Contradiction

Suppose all concepts are naturally object-oriented
Classes In the Real World

Ape

Orangutan

Gorilla

Human

Bonobo

Chimp
Classes In the Real World

gorilla.pluck(leaves)

human.sleep(5)

chimp.eat(fruit)
Classes In Finance

- Option
  - European
  - American
  - Index
  - Bermudan
Classes In Finance

- option.underlier()
- option.gamma()
- portfolio.var(20, 0.95)
Classes In Math

Number
- Integer
- Rational
- Irrational
- Complex
- Quaternion
Classes In Math

1. add(4)
2. 16.log(2)
3. 13.next_fibonacci()
Contradiction

Math is not object-oriented
<table>
<thead>
<tr>
<th>Expression</th>
<th>Simplified</th>
</tr>
</thead>
<tbody>
<tr>
<td>add(4)</td>
<td>1 + 4</td>
</tr>
<tr>
<td>log(2)</td>
<td>log₂ 16</td>
</tr>
<tr>
<td>next_fibonacci()</td>
<td>f(n) = f(n-1) + f(n-2)</td>
</tr>
</tbody>
</table>
Finance Is Functional

\[ \text{Cov}(X, Y) \]

\[ \min w^T \Sigma w \]
Finance Is Functional

\[ \text{Cov}(X, Y) \]

\[ \min w^T \sum w \]

\[ \text{w.transpose().mult(\sum).mult(w).min()} \]
R Is Functional

\[
\begin{align*}
\text{fib.1} & \quad \%\text{when}\% \quad (n \ %\text{in}\% \ c(0,1)) \\
\text{fib.1} & \quad \leftarrow \quad \text{function}(n) \ 1 \\
\text{fib.2} & \quad \%\text{when}\% \quad (n > 1) \\
\text{fib.2} & \quad \leftarrow \quad \text{function}(n) \\
& \quad \quad \text{fib}(n-1) + \text{fib}(n-2) \\
\end{align*}
\]

> fib(6)
[1] 13
R Is Functional

coupon.pct %when% (bond %isa% Bond && bond$coupon < 1)
coupon.pct <- function(bond)
  100 * bond$coupon / bond$freq

coupon.dv %when% (bond %isa% Bond)
coupon.dv %must% (result > 0)
coupon.dv <- function(bond)
  bond$coupon / bond$freq

> b <- create(Bond,coupon=.035,freq=2)
> coupon(b)
[1] 1.5
install.packages("futile.paradigm")