Functional Reporting

Edward Kmett

Overview

1. Who We Are
2. Getting FP in the Door
   Monoids, Reducers, Iteratees, Performance Attribution
3. Ermine
   “Haskell with Row Types”
4. Reporting
   EDSL Inception and JMacro RPC
5. Open Source

Overview

1. Who We Are
2. Getting FP in the Door
   Monoids, Reducers, Iteratees, Performance Attribution
3. Ermine
   “Haskell with Row Types”
4. Reporting
   EDSL Inception and JMacro RPC
5. Open Source

Who We Are

Over 4,500 customers including

- Investment Management Firms
- Private Equity Funds
- Investment Banks
- Advisory Firms
- Corporations
- Universities

Who We Are

Products Include

- S&P Capital IQ Platform
- ClariFI
- AlphaWorks
- Compustat

Overview

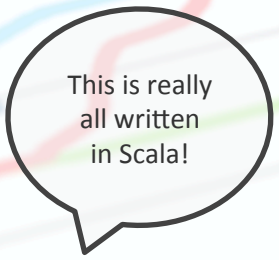
1. Who We Are
2. **Getting FP in the Door**
   Monoids, Reducers, Iteratees, Performance Attribution
3. Ermine
   "Haskell with Row Types"
4. Reporting
   EDSL Inception and JMacro RPC
5. Open Source

Getting FP in the Door

- Monoids
- Reducers
- Performance Attribution
- Factor Backtesting

Monoids

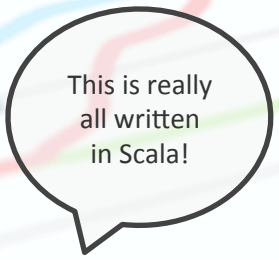
```
class Monoid m where
  mappend :: m -> m -> m
  mempty  :: m

newtype Sum a = Sum { getSum :: a }
instance Num a => Monoid (Sum a) where
  mempty = Sum 0
  mappend (Sum m) (Sum n) = Sum (m + n)
```



Monoids

```
class Monoid m where
  mappend :: m -> m -> m
  mempty  :: m
```

```
>>> foldMap Sum [1,2,3]
6
```



Reducers

```
class Monoid m where
  mappend :: m -> m -> m
  mempty  :: m

data Reducer a b =
  forall m. Monoid m => R (a -> m) (m -> b)

reduce :: Reducer a b -> [a] -> b
reduce (R am mb) xs = mb (foldMap am xs)
```

In practice you probably want an efficient `cons` and/or `snoc` operation as well.

Simultaneous Reduction

```
instance (Monoid a, Monoid b) => Monoid (a,b)

instance Applicative (Reducer a) where
  pure a = R (\_ -> ()) (\() -> a)
  R am mf <*> R an nx = R (am &&& an) $
     \(m,n) -> mf m $ nx n

mean = (/) <$> sum <*> length
```

Simultaneous Reduction

```
instance Num b => Num (Reducer a b)
instance Fractional b => Fractional (Reducer a b)

mean = sum / length
```

Performance Attribution

OLD AND BUSTED

- Implemented in Java
- Needed Full Dataset in Memory
- Hard to Extend
- Results are Serialized Objects

NEW HOTNESS

- Implemented in Scala
- Runs in Constant Memory
- Easily Extended
- Drastic Speed Improvements
- Results Flattened via Combinators to Database

Performance Attribution

OLD AND BUSTED

- Implemented in Java
- Needed Full Dataset in Memory
- Hard to Extend
- Results are Serialized Objects

NEW HOTNESS

- Implemented in Scala
- Runs in Constant Memory
- Easily Extended
- Drastic Speed Improvements
- Results Flattened via Combinators to Database

GLOSSED OVER

- Multi-Pass Algorithms
- Iteratees
- Caching
- Aggregation by Sectors
- Missing Data

Portfolio Attribution Report

Summary report for long/short portfolios

this report has been exported from a performance attribution workflow run in ModelStation from Clarifi

REPORT INFORMATION

General information about the Portfolio Attribution Report run in ModelStation and exported to create this workbook.

Portfolio Name	Dow Long / Short
Benchmark	4 stock
Returns Attribution	Beginning of period holdings
Risk Model	Capital IQ Risk Model
Analysis Period	110/11/30 – 111/2/30

RETURNS

These statistics summarize the characteristics of the returns (losses) of the portfolio over the duration of the analysis perio
ModelStation analysis, benchmark-relative statistics are also provided for the period.

BRINSON ATTRIBUTION

These charts and tables show the cumulative attribution of the Brinson effects. Periodic data is linked using Carino smoothi

Performance Attribution Demo

Overview

1. Who We Are
2. Getting FP in the Door
   Monoids, Reducers, Iteratees, Performance Attribution
3. Ermine
   "Haskell with Row Types"
4. Reporting
   EDSL Inception and JMacro RPC
5. Open Source

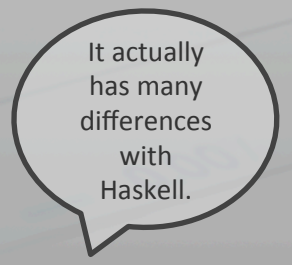
Ermine

Ermine

"Haskell with Row Types"

FEATURES

- Not Scala
- Portable Core that can run on the JVM
- Implementations in Scala and Haskell
- Strong Type System
  - Novel Row Types
  - Polymorphic and Constraint Kinds
  - Rank-N Types via a derivative of HMF
- Fits our Domain
  - Built-In Database Support
  - Can Be Exposed to End Users
  - Prototype Structured Code Editor
  - Full Control Over Error Messages



Row Polymorphism

PREVIOUS APPROACHES

Commonly row polymorphism is modeled via has, lacks, and/or subsumes constraints.

```
cons_l : ∀(a : *) (r : ρ). (r / l) ⇒ α → [..r] → {l : α|r}
tail_l : ∀(α : *) (r : ρ). (r / l) ⇒ {l : α|r} → {r}
join : ∀(r : ρ) (s : ρ). [..r] → [..s] → [.. r || s]
```

This is easily checked, but now **inference** flows unidirectionally through join.

Row Polymorphism

IN ERMINE

We use a single constraint type: "can be partitioned into"

```
a ← (b,c)
```

says the fields in row type a can be partitioned into **disjoint** sets of fields b and c.

```
join :
  ( d ← (a,b)
  , e ← (b,c)
  , f ← (a,b,c)
  ) ⇒ [..d] → [..e] → [..f]
```

Row Polymorphism

IN ERMINE

We can have existentials in our constraint types. E.g.

```
forget : MonadState s m ⇒ m a →  m a
forget m = do s ← get; a ← m; put s; return a
```

has type parameters that don't occur on the right hand side of the (⇒) determined by functional dependencies. We could give this type:

```
forget : (∃s. MonadState s m) ⇒ m a -> m a
```

Using this we can model "Has" and "Lacks" or "Disjoint" via type aliases!

```
type Has a b = ∃c. a ← (b, c)
type a | b   = ∃c. c ← (a, b)
```

Overview

1. Who We Are
2. Getting FP in the Door
   Monoids, Reducers, Iteratees, Performance Attribution
3. Ermine
   "Haskell with Row Types"
4. Reporting
   EDSL Inception and JMacro RPC
5. Open Source

Reporting

A Language in our Language

S&P CAPITAL IQ

Portfolio Information

Portfolio Name	big portfolio	Date Range	4/30/2012 to 4/30/2013
Portfolio ID	-	Currency	US Dollar (USD)
Benchmark	Russell Top 50 Index (^RTF)	Grouping	GICS Industry
Risk Model	Global Fundamental Short Term	Exclude Cash	No

Portfolio Overview

	Port	Bench	Active
Total Return	28.98	14.37	14.61
Realized Risk	16.29	12.28	10.59
Latest Predicted Risk	12.99	8.69	9.44
-Factor Risk	9.79	8.33	4.19
-Stock Specific Risk	8.54	2.47	8.46
Portfolio Value($mm)	0.15	-	-
# of Securities	17	52	-
Annualized Total Return	28.98	14.37	14.61

Portfolio Return Chart

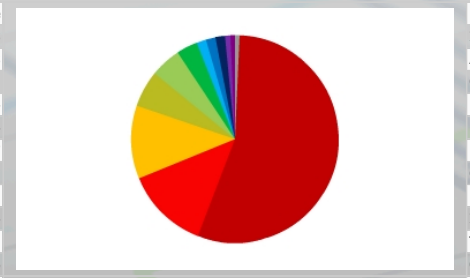
Portfolio Characteristics

	Port	Bench
Market Capitalization	204,448.67	192,445.11
P/E	11.56	15.55
P/BV	2.80	3.88
P/Sales	3.61	2.59
Div Yield	2.79	2.67
Return on Equity %	16.79	32.32
Total Debt/Equity	102.27	624.19
Forward P/E	13.01	13.63
EPS 3 Yr CAGR %	28.36	14.04

Industry Weight

	Port	Bench	Active
Internet Software and Ser...	55.11	3.23	51.88
Automobiles	12.98	0.00	12.98
Capital Markets	11.40	0.00	11.40
Pharmaceuticals	5.70	10.54	-4.84
Food Products	4.78	1.24	3.54
Diversified Financial Serv...	3.28	6.55	-3.27
Commercial Services and...	1.51	0.00	1.51
Electric Utilities	1.50	0.00	1.50
Industrial Conglomerates	1.49	4.43	-2.94
Trading Companies and D...	0.83	0.00	0.83
Biotechnology	0.71	1.16	-0.46
Other	0.73	72.85	-72.12

Industry Weight Chart



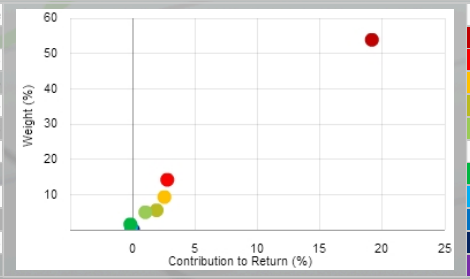
Industry Attribution

	Alloc Eff	Selec Eff	Inter Eff	Total Eff
Internet Software and...	8.54	0.12	2.23	10.90
Automobiles	1.78	-0.05	-1.15	0.58
Capital Markets	1.59	0.02	0.18	1.79
Pharmaceuticals	-0.65	0.33	-0.17	-0.49
Food Products	0.41	-0.04	-0.20	0.17
Diversified Financial ...	-0.54	-0.82	0.33	-1.03
Commercial Services ...	-0.34	0.00	0.00	-0.34
Electric Utilities	-0.01	0.00	0.00	-0.01
Industrial Conglomer...	-0.11	-0.02	0.01	-0.12
Trading Companies a...	-0.28	0.00	0.00	-0.28
Other	3.80	1.02	-1.37	3.45
Total	14.18	0.56	-0.13	14.61

Security Weight

Top 10	Port	Bench	Active
Google Inc. (NasdaqGS:GO...	55.11	3.06	52.05
Volkswagen AG (DB:VOW)	12.98	0.00	12.98
The Goldman Sachs Group, ...	9.76	0.00	9.76
Johnson & Johnson (NYSE:J...	5.70	3.36	2.34
Nestlé S.A. (SWX:NESN)	4.78	0.00	4.78
JPMorgan Chase & Co. (NY...	3.28	2.68	0.60
ABM Industries Incorporat...	1.51	0.00	1.51
Electricite de France SA (E...	1.50	0.00	1.50
General Electric Company (...	1.49	3.39	-1.90
Gluskin Sheff + Associates,...	1.18	0.00	1.18
Total	97.28	12.49	84.79

Security Contribution Chart



Security Contribution

Top 5	Avg Wgt	Return	Contr
Google Inc. (NasdaqGS:G...	53.86	36.33	19.16
Volkswagen AG (DB:VOW)	14.17	16.33	2.78
The Goldman Sachs Grou...	9.33	28.98	2.53
Johnson & Johnson (NYS...	5.53	35.63	1.92
Nestlé S.A. (SWX:NESN)	5.01	20.34	1.02
Bottom 5	**Avg Wgt**	**Return**	**Contr**
ABM Industries Incorpor...	1.59	-0.25	-0.18
Sumitomo Corporation (T...	1.03	-9.04	-0.14
Gefran SpA (BIT:GE)	0.28	-24.41	-0.12
Granville Pacific Capital ...	0.01	-3.91	0.00
Advanced Power Compon...	0.02	241.45	0.03

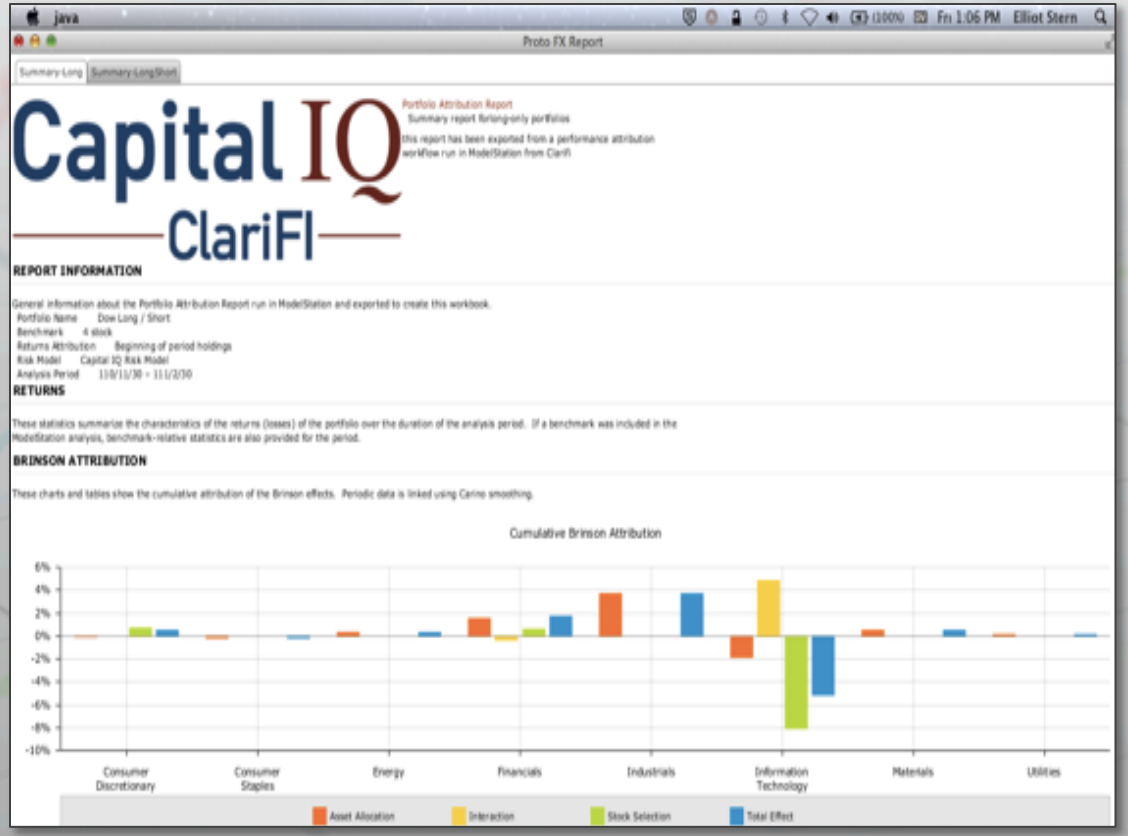
```
contributionSnapshotReport rid = dropdown
 (remoteChoice dateRangePresentation (reportingRanges rid)) $
 \ dateRangeDropdown dateRangeSelector ->
  topAndCenter
    (hugL $ hflow ["Selected Period: ", dateRangeDropdown])
    (using dateRangeSelector (\ dr -> topAndCenter
      (contributionSnapshotSummary rid (dr ! dateRangeId))
      (tabbed [
        ("Net",   contributionSnapshotReportContent rid (dr ! dateRangeId) LSNNet),
        ("Long",  contributionSnapshotReportContent rid (dr ! dateRangeId) LSNLong),
        ("Short", contributionSnapshotReportContent rid (dr ! dateRangeId) LSNShort)
      ]))
    )
```

Reporting Example

One Report
Multiple Interpreters

The same report specification can run under wildly different interpreters. E.g.

- HTML + AJAX
- Excel
- PDF
- Java Swing



Overview

1. Who We Are
2. Getting FP in the Door
   Monoids, Reducers, Iteratees, Performance Attribution
3. Ermine
   “Haskell with Row Types”
4. Reporting
   EDSL Inception and JMacro RPC
5. Open Source

Open Source

SCALA VERSION

- https://github.com/ermine-language/ermine-legacy

HASKELL VERSION

- https://github.com/ermine-language/ermine

NEW CORE INTERPRETER

- https://github.com/ermine-language/ermine-scala-core

JMACRO-RPC

- http://patch-tag.com/r/gershomb/jmacro-rpc

Questions?