

# Icicle

Write once, loop once

Amos Robinson, Ambiata (and UNSW)

Huw Campbell, Mark Hibberd, Tran Ma, Jacob Stanley

# Company data

Code	Date	Open	Close
ABC	2015-07-01	19.00	19.50
ABC	2015-06-01	20.00	20.50
ABC	2015-05-01	21.00	21.50
.	.	.	.
IAG	2015-11-02	5.60	5.55
IAG	2015-10-02	4.80	4.85
IAG	2015-09-01	5.05	5.05
.	.	.	.
DEF	2015-07-01	10.00	10.00
DEF	2015-06-01	9.00	9.00
DEF	2015-05-01	8.00	8.00

# Minimum

Code	Date	Open	Close	min open
ABC	2015-07-01	19.00	19.50	19.00
ABC	2015-06-01	20.00	20.50	
ABC	2015-05-01	21.00	21.50	
.	.	.	.	.
IAG	2015-11-02	5.60	5.55	4.80
IAG	2015-10-02	4.80	4.85	
IAG	2015-09-01	5.05	5.05	
.	.	.	.	.
DEF	2015-07-01	10.00	10.00	8.00
DEF	2015-06-01	9.00	9.00	
DEF	2015-05-01	8.00	8.00	

# Mean gap

Code	Date	Open	Close	mean (close-open)
ABC	2015-07-01	19.00	19.50	0.50
ABC	2015-06-01	20.00	20.50	
ABC	2015-05-01	21.00	21.50	
.	.	.	.	.
IAG	2015-11-02	5.60	5.55	0.00
IAG	2015-10-02	4.80	4.85	
IAG	2015-09-01	5.05	5.05	
.	.	.	.	.
DEF	2015-07-01	10.00	10.00	0.00
DEF	2015-06-01	9.00	9.00	
DEF	2015-05-01	8.00	8.00	

# Big data

Code	Date	Open	Close
ABC	2015-07-01	19.00	19.50
ABC	2015-06-01	20.00	20.50
ABC	2015-05-01	21.00	21.50
.	.	.	.
IAG	2015-11-02	5.60	5.55
IAG	2015-10-02	4.80	4.85
IAG	2015-09-01	5.05	5.05
.	.	.	.
DEF	2015-07-01	10.00	10.00
DEF	2015-06-01	9.00	9.00
DEF	2015-05-01	8.00	8.00

(daily data for ASX is ~500mb CSV, but let's pretend)

# Many queries

max close	min close	min open	mean (close-open)
-----------	-----------	----------	-------------------

21.50

19.50

19.00

0.50

5.55

4.85

4.80

0.00

10.00

8.00

8.00

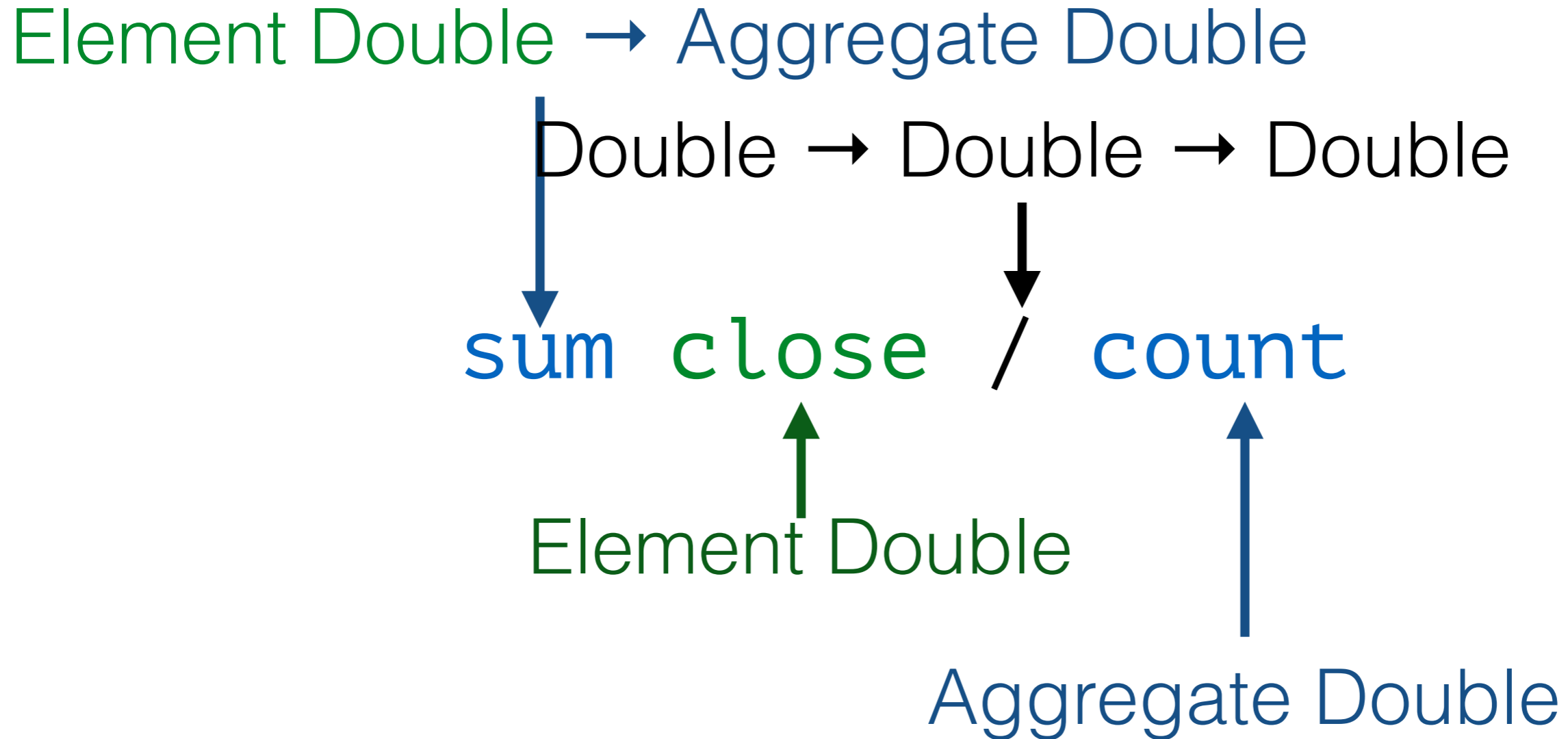
0.00

(they do get more interesting...)

# Icicle

A single-pass query language

# Mean





# Sum

sum v  
= fold s = 0  
: s + v  
~> s

# Count

```
count = sum 1
```

# Filter

~> filter close > open  
count

# Single pass restriction

~> filter count [open] > [mean open]

# Query plan (Core)

Before

$b = \text{Exp}$

Streams

$s = \text{Source} \quad | \quad \text{Map} \quad f \quad s$   
 $\quad | \quad \text{Filter} \quad p \quad s$

Reduces

$r = \text{Fold} \quad k \quad z \quad s$

After

$a = \text{Exp}$

Outputs  $a \dots$

# Sum open

Before

Streams

$o' = \text{Map open Source}$

Reduces

$s = \text{Fold } (+) \emptyset o'$

After

Outputs  $s$

$\text{sum open}$

$= \text{fold } s$

$= \emptyset$

$: s + \text{open}$

$\sim \>$

$s$

# Mean open

Before

Streams

$o' = \text{Map } \text{open} \text{ Source}$

$cs = \text{Map } 1 \text{ Source}$

Reduces

$s = \text{Fold } (+) \ 0 \ o'$

$c = \text{Fold } (+) \ 0 \ cs$

After

$r = s / c$

Outputs  $r$

$\text{mean } \text{open}$   
 $= \text{fold } s$   
 $= 0$   
 $: s + \text{open}$   
 $\sim > \text{fold } c$   
 $= 0$   
 $: c + 1$   
 $\sim > s / c$

# Fusion

Before

Streams

`o1 = Map open Source`

Reduces

`s1 = Fold (+) 0 o1`

After

Outputs `s1`

Before

Streams

`o2 = Map open Source`  
`cs2 = Map 1 Source`

Reduces

`s2 = Fold (+) 0 o2`  
`c2 = Fold (+) 0 cs2`

After

`r2 = s2 / c2`  
Outputs `r2`



# Fusion

Before

Streams

Reduces

After

Outputs

Before

Streams

```
o2 = Map open Source  
cs2 = Map 1 Source
```

Reduces

```
s2 = Fold (+) 0 o2  
c2 = Fold (+) 0 cs2
```

After

```
r2 = s2 / c2  
Outputs r2 s1
```

# Code generation

# Code generation

Before

Streams `foreach fact in facts`

```
o' = Map open Source {  
cs = Map 1 Source
```

Reduces

```
s = Fold (+) 0 o'  
c = Fold (+) 0 cs
```

After

```
r = s / c
```

Outputs `r`

}

# Code generation

Before

Streams

```
o' = Map open Source
cs = Map 1 Source
```

```
foreach fact in facts
```

```
{
```

```
o' = fact.open
```

Reduces

```
s = Fold (+) 0 o'
c = Fold (+) 0 cs
```

After

```
r = s / c
```

Outputs r

```
}
```

# Code generation

Before

Streams

```
o' = Map open Source  
cs = Map 1 Source
```

```
foreach fact in facts
```

```
{  
  o' = fact.open
```

Reduces

```
s = Fold (+) 0 o'  
c = Fold (+) 0 cs
```

After

```
r = s / c
```

Outputs r

```
}
```

# Code generation

Before

```
s = 0
```

Streams

```
o' = Map open Source  
cs = Map 1 Source
```

```
foreach fact in facts
```

```
{  
  o' = fact.open
```

```
  s = s + o'
```

Reduces

```
s = Fold (+) 0 o'  
c = Fold (+) 0 cs
```

After

```
r = s / c
```

Outputs r

```
}
```

# Code generation

Before

```
s = 0
```

Streams

```
o' = Map open Source  
cs = Map 1 Source
```

```
foreach fact in facts
```

```
{  
  o' = fact.open  
  s = s + o'
```

Reduces

```
s = Fold (+) 0 o'  
c = Fold (+) 0 cs
```

```
cs = 1
```

After

```
r = s / c
```

Outputs r

```
}
```

# Code generation

Before

$s = 0$

Streams

```
o' = Map open Source  
cs = Map 1 Source
```

```
foreach fact in facts
```

```
{  
  o' = fact.open  
  s = s + o'
```

Reduces

```
s = Fold (+) 0 o'  
c = Fold (+) 0 cs
```

```
cs = 1
```

After

```
r = s / c
```

Outputs r

```
}
```



# Code generation

Before

```
s = 0
```

```
c = 0
```

Streams

```
foreach fact in facts
```

```
o' = Map open Source {
```

```
cs = Map 1 Source
```

```
o' = fact.open
```

```
s = s + o'
```

Reduces

```
s = Fold (+) 0 o'
```

```
cs = 1
```

```
c = Fold (+) 0 cs
```

```
c = c + cs
```

After

```
r = s / c
```

Outputs r

```
}
```

# Code generation

Before

$s = 0$

$c = 0$

Streams

foreach fact in facts

$o' = \text{Map open Source}$

$cs = \text{Map 1 Source}$

{  
 $o' = \text{fact.open}$

$s = s + o'$

Reduces

$s = \text{Fold } (+) \ 0 \ o'$

$c = \text{Fold } (+) \ 0 \ cs$

$cs = 1$

$c = c + cs$

After

$r = s / c$

}

$r = s / c$

Outputs  $r$

# Code generation

Before

$s = 0$

$c = 0$

Streams

foreach fact in facts

$o' = \text{Map open Source}$

$cs = \text{Map 1 Source}$

{  
 $o' = \text{fact.open}$

$s = s + o'$

Reduces

$s = \text{Fold } (+) \ 0 \ o'$

$c = \text{Fold } (+) \ 0 \ cs$

$cs = 1$

$c = c + cs$

After

$r = s / c$

}

$r = s / c$

Outputs  $r$

DEMO DEMO DEMO DEMO

DEMO DEMO DEMO DEMO

DEMO DEMO DEMO DEMO

DEMO DEMO DEMO DEMO

DEMO DEMO DEMO DEMO

DEMO DEMO DEMO DEMO

DEMO DEMO DEMO DEMO

DEMO DEMO DEMO DEMO