# Volatile Stocks

Ken Hebel

# What I wanted to do...

- *Find the most volatile stocks over a time period*
  *Given*
    *2 dates || an end date and a time period*

- *Find the moving average of those stocks (and others?) given the same kind of input.*

- *Integrate into the current Akka web app.*
    *using the current "go" button*

# Find the most volatile stocks over a time period

```
var map1 = function () {
  var delta = 0;
  delta = Math.abs(this.close - this.open)
  emit(this.stock_symbol, {delta: delta});
};
```

Nothing special about the reduce function

Create array of the most volatile stock symbols for later use...

# Find the moving averages of those stocks

... Nope!

kind of.

# Integrate into web-app

Step 1: Attempt to understand how to web-app works.

Step 2: ???????

Step 3: hope you don't fail because you couldn't deliver.

App.js => RestfulDataPublisher.getAllDataFor =>
That returns some kind of JSON Map...

MongoDBDataStore == query????

# Simple moving average

```
var map2 = function () {
 var windowSize = 25;
 var array = db.A_prices.find({stock_symbol: this.stock_symbol, date: {$gte: this.date}}).
limit(windowSize);
 var answer = array.toArray();
  emit(this.date, {symbol: this.stock_symbol, avg: answer});
}

var reduce2 = function (key, values) {
  var sum = 0;
  for (var n = 0; n<values[0].avg.length; n++)
sum += values[0].avg[n].close;
  var answer = sum / values[0].avg.length;
  return {symbol: values[0].avg[0].stock_symbol, avg: answer};
}
for (var k = 0; k<5; k++){
  var result2 = db.A_prices.mapReduce(map2, reduce2, { query: {stock_symbol:
symbolArray[k], date: {$gte: beginDate, $lte: endDate}}});
```

# Plotting!

Q/A?