simple-map.json

A simple map-only job that returns the entire data set.

```
{"inputs":"goog",
  "query":[{"map":{"language":"javascript",
                   "name":"Riak.mapValuesJson",
                   "keep":true}}
          ]
}
```

map-high.json

A map-reduce job that returns the maximum high sell value in the first week of January.

```
{"inputs":[["goog","2010-01-04"],
           ["goog","2010-01-05"],
           ["goog","2010-01-06"],
           ["goog","2010-01-07"],
           ["goog","2010-01-08"]],
 "query":[{"map":{"language":"javascript",
                  "source":"function(value,keyData,arg){ var data =
Riak.mapValuesJson(value)[0]; return [data.High];}"
                 }},
          {"reduce":{"language":"javascript","name":"Riak.reduceMax","keep":true}}]
}
```

map-highs-by-month.json

A more complicated map-reduce job that collects the max high by month.

```
{
    "inputs":"goog",
    "query":[
        {"map":{"language":"javascript", "source":"function(value, keyData, arg){ var
data = Riak.mapValuesJson(value)[0]; var month = value.key.split('-
').slice(0,2).join('-'); var obj = {}; obj[month] = data.High; return [obj];}"}},
    {"reduce":{"language":"javascript", "source":"function(values, arg){ return [
values.reduce(function(acc, item){ for(var month in item){ if(acc[month]) { acc[month]
= (acc[month] < item[month]) ? item[month] : acc[month]; } else { acc[month] =
item[month]; } } return acc; })];}", "keep":true}}
]
}
```

first-week.json

A simple map-only job that returns the values for the first week of January 2010.

```
{"inputs":[["goog","2010-01-04"],
           ["goog","2010-01-05"],
           ["goog","2010-01-06"],
           ["goog","2010-01-07"],
           ["goog","2010-01-08"]],
   "query":[{"map":{"language":"javascript","name":"Riak.mapValuesJson","keep":true}}]
}
```

Map: find the days where the high was over $600.00

Phase Function

```
function(value, keyData, arg) {
  var data = Riak.mapValuesJson(value)[0];
  if(data.High && data.High > 600.00)
    return [value.key];
  else
    return [];
}
```

Complete Job

```
{"inputs":"goog",
 "query":[{"map":{"language":"javascript",
                  "source":"function(value, keyData, arg) { var data =
Riak.mapValuesJson(value)[0]; if(data.High && parseFloat(data.High) > 600.00)
return [value.key]; else return [];}",
                  "keep":true}}]
}
```

Map: find the days where the close is lower than open

Phase Function

```
function(value, keyData, arg) {
  var data = Riak.mapValuesJson(value)[0];
  if(data.Close < data.Open)
    return [value.key];
  else
    return [];
}
```

Complete Job

```
{"inputs":"goog",
 "query":[{"map":{"language":"javascript",
                  "source":"function(value, keyData, arg) { var data =
Riak.mapValuesJson(value)[0]; if(data.Close < data.Open) return [value.key]; else
return [];}",
                  "keep":true}}]
}
```

Map and Reduce: find the maximum daily variance in price by month

Phase functions

```
/* Map function to compute the daily variance and key it by the month */
function(value, keyData, arg){
  var data = Riak.mapValuesJson(value)[0];
  var month = value.key.split('-').slice(0,2).join('-');
  var obj = {};
  obj[month] = data.High - data.Low;
  return [ obj ];
}

/* Reduce function to find the maximum variance per month */
function(values, arg){
  return [ values.reduce(function(acc, item){
              for(var month in item){
                  if(acc[month]) { acc[month] = (acc[month] < item[month]) ?
item[month] : acc[month]; }
                  else { acc[month] = item[month]; }
              }
              return acc;
             })
         ];
}
```

Complete Job

```
{"inputs":"goog",
 "query":[{"map":{"language":"javascript",
                  "source":"function(value, keyData, arg){ var data =
Riak.mapValuesJson(value)[0]; var month = value.key.split('-').slice(0,2).join('-
'); var obj = {}; obj[month] = data.High - data.Low; return [ obj ];}"}},
         {"reduce":{"language":"javascript",
                    "source":"function(values, arg){ return [
values.reduce(function(acc, item){ for(var month in item){ if(acc[month]) {
acc[month] = (acc[month] < item[month]) ? item[month] : acc[month]; } else {
acc[month] = item[month]; } } return acc;  }) ];}",
                    "keep":true}}
         ]
}
```