```
/* global dc , d3 , crossfilter , $ */

// Create plot variable names
var cuisine_chart = dc.pieChart( '#recipe-cuisine-chart' );
var country_of_origin_chart = dc.pieChart( '#recipe-country-of-origin-chart' );
var servings_chart = dc.pieChart( '#recipe-servings-chart' );
var dietary_chart = dc.rowChart( '#recipe-dietary-chart' );
var allergen_chart = dc.rowChart( '#recipe-allergen-chart' );
var difficulty_chart = dc.rowChart( '#recipe-difficulty-chart' );
var main_ingredient_chart = dc.barChart( '#recipe-main-ingredient-chart' );
var total_time_chart = dc.barChart( '#recipe-total-time-chart' );

// check if plot data from index.html is empty
if ( $( '#data' ).text() != '' ) {
  var data = JSON.parse( $( '#data' ).text() ); // import plot data from index.htm
  var ndx = crossfilter( data ); // create crossfilter from data

  // create pie chart of cuisine data
  var cuisine_dimension = ndx.dimension( function(d) { return d.recipeCuisine; } )
  var cuisine_group = cuisine_dimension.group();
  cuisine_chart
    .height( 250 )
    .width( 250 )
    .dimension( cuisine_dimension )
    .group( cuisine_group )
    .title( function(d) { return 'Number of ' + d.key + ' recipes: ' + d.value; }

  // create pie chart of country of origin data
  var country_of_origin_dimension = ndx.dimension( function(d) { return d.recipeCo
  var country_of_origin_group = country_of_origin_dimension.group();
  country_of_origin_chart
    .height( 250 )
    .width( 250 )
    .dimension( country_of_origin_dimension )
    .group( country_of_origin_group )
    .title( function(d) { return 'Number of recipes from ' + d.key + ': ' + d.valu

  // create pie chart of number of servings data
  var servings_dimension = ndx.dimension( function(d) { return d.recipeServings; }
  var servings_group = servings_dimension.group();
  servings_chart
    .width( 250 )
    .height( 250 )
    .dimension( servings_dimension )
    .group( servings_group )
    .label( function(d) { return d.key + ' servings'; } )
```

```
47          .title( function(d) { return 'Number of recipes with ' + d.key + ' servings: '
48
49      //create row chart of dietary data
50      var dietary_dimension = ndx.dimension( function(d) { return d.recipeDietary; },
51      var dietary_group = dietary_dimension.group();
52      dietary_chart
53        .renderLabel( true )
54        .height( 250 )
55        .width( 250 )
56        .dimension( dietary_dimension )
57        .group( dietary_group )
58        .cap( 10 )
59        .title( function(d) { return 'Number of recipes which are ' + d.key + ': ' + d
60        .ordering( function(d) { return -d.value; } )
61        .xAxis()
62        .ticks( 3 );
63
64      //create row chart of allergen data
65      var allergy_dimension = ndx.dimension( function(d) { return d.recipeAllergen; },
66      var allergy_group = allergy_dimension.group();
67      allergen_chart
68        .renderLabel( true )
69        .height( 250 )
70        .width( 250 )
71        .dimension( allergy_dimension )
72        .group( allergy_group )
73        .cap( 10 )
74        .title( function(d) { return 'Number of recipes which contain ' + d.key + ': '
75        .ordering( function(d) { return -d.value; } )
76        .xAxis()
77        .ticks( 3 );
78
79      //create row chart of difficulty data
80      var difficulty_dimension = ndx.dimension( function(d) { return d.recipeDifficulty
81      var difficulty_group = difficulty_dimension.group();
82      difficulty_chart
83        .height( 250 )
84        .width( 250 )
85        .dimension( difficulty_dimension )
86        .group( difficulty_group )
87        .cap( 10 )
88        .title( function(d) { return 'Number of recipes with difficulty ' + d.key + ':
89        .ordering( function(d) { return -d.value; } )
90        .xAxis()
91        .ticks( 3 );
92
93      //create bar chart of main ingredient data
94      var main_ingredient_dimension = ndx.dimension( dc.pluck( 'recipeMainIngredient'
```

CONFIGURE

Metrics

There are 19 functions in this file.

Function with the largest signature take 1 arguments, while the median is 1.

Largest function has 8 statements in it, while the median is 1.

The most complex function has a cyclomatic complexity value of 6 while the median is 1.

JSHint

version 2.10.2
(https://github.com
/jshint/jshint/releases
/tag/2.10.2)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

```
 95      var main_ingredient_group = main_ingredient_dimension.group();
 96      main_ingredient_chart
 97        .height( 250 )
 98        .width( 500 )
 99        .margins( { top: 10, right: 50, bottom: 50, left: 50 } )
100        .dimension( main_ingredient_dimension )
101        .group( main_ingredient_group )
102        .title( function(d) { return 'Number of recipes with ' + d.key + ' as the main
103        .transitionDuration( 500 )
104        .x( d3.scale.ordinal() )
105        .xUnits( dc.units.ordinal )
106        .xAxisLabel( 'Main Ingredient' )
107        .yAxisLabel( 'Number of recipes' )
108        .yAxis()
109        .ticks( 5 );
110
111      //create bar chart of total time data
112      var total_time_dimension = ndx.dimension( function(d) {
113        var time = d.recipePreparationTime + d.recipeCookingTime;
114        if ( time < 30 ) { return '< 30'; }
115        else if ( time >= 30 && time < 60 ) { return '< 60'; }
116        else if ( time >= 60 && time < 90 ) { return '< 90'; }
117        else if ( time >= 90 && time < 120 ) { return '< 120'; }
118        else if ( time >= 120 && time < 150 ) { return '< 150'; }
119        else { return '> 150'; }
120      });
121      var total_time_group = total_time_dimension.group();
122      total_time_chart
123        .height( 250 )
124        .width( 500 )
125        .margins( { top: 10, right: 50, bottom: 50, left: 50 } )
126        .dimension( total_time_dimension )
127        .group( total_time_group )
128        .title( function(d) { return 'Number of recipes with total time ' + d.key + ' i
129        .transitionDuration( 500 )
130        .x( d3.scale.ordinal() )
131        .xUnits( dc.units.ordinal )
132        .xAxisLabel( 'Total time (minutes)' )
133        .yAxisLabel( 'Number of recipes' )
134        .yAxis()
135        .ticks( 5 );
136
137      // render all charts on page
138      dc.renderAll();
139    }
140
```

CONFIGURE

Metrics

There are 19 functions in this file.

Function with the largest signature take 1 arguments, while the median is 1.

Largest function has 8 statements in it, while the median is 1.

The most complex function has a cyclomatic complexity value of 6 while the median is 1.

version 2.10.2
(https://github.com
/jshint/jshint/releases
/tag/2.10.2)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

CONFIGURE

Metrics

There are 19 functions in this file.

Function with the largest signature take 1 arguments, while the median is 1.

Largest function has 8 statements in it, while the median is 1.

The most complex function has a cyclomatic complexity value of 6 while the median is 1.

JSHint

version 2.10.2
(https://github.com
/jshint/jshint/releases
/tag/2.10.2)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)