```
// Hello.
//
// This is JSHint, a tool that helps to detect errors and potential
// problems in your JavaScript code.
//
// To start, simply enter some JavaScript anywhere on this page. Your
// report will appear on the right side.
//
// Additionally, you can toggle specific options in the Configure
// menu.

//Identify global variables
/* global dc, d3, crossfilter, $, topFunction */


//Event actions
$(document).ready(function() {

    $('#initial-search-date').datepicker({ //select date to start search
        dateFormat: 'yy-mm-dd'
    });

    $('#initial-search-date').focus(function() {
        $('.ui-datepicker').addClass('calendar-background'); //add class to date-p
    });

    $('button.data').click(function() {
        var input_date =  document.getElementById('initial-search-date').value;
        var  date_validaton = Date.parse(input_date);

            if (date_validaton > 0) {
                var button_id = this.id.toString(); //get the id of the search but

                if ($('#data-output').hasClass("data-hidden")) {
                    var data_output_state = 'false';
                }
                else {data_output_state = 'true';}

                neo_search_period(button_id, data_output_state); //call search per
            }
            else {
                document.getElementById('initial-search-date').value = 'Please ent
                alert("An invalid date has been entered");
            }
        });
```

CONFIGURE

JSHint version 2.9.6 (https://github.com/jshint/jshint/releases/tag/2.9.6)

About (/about)
Documentation (/docs)
Install (/install)
Contribute (/contribute)
Blog (/blog)

## Metrics

There are 61 functions in this file.

Function with the largest signature take 3 arguments, while the median is 1.

Largest function has 19 statements in it, while the median is 1.

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

## 29 warnings

- 37 'data_output_state' used out of scope.
- 39 'data_output_state' used out of scope.
- 47 Missing semicolon.
- 48 Missing semicolon.
- 91 Expected an assignment or function call and instead saw an expression.
- 105 Expected an assignment or function call and instead saw an expression.
- 167 ['name'] is better written in dot notation.
- 168 ['nasa_jpl_url'] is better written in dot notation.
- 169 ['close_approach_data'] is better written in dot notation.
- 169 ['close_approach_date'] is better written in dot notation.
- 170 ['absolute_magnitude_h'] is better written in dot notation.
- 171 ['estimated_diameter'] is better written in dot notation.
- 171 ['kilometers'] is better written in dot notation.
- 171 ['estimated_diameter_max'] is better written in dot notation.

```
47          $('.link').mouseenter(function() { $(this).animate({ "color": "#eb5d5d" }, 800
48              .mouseleave(function() { $(this).animate({ "color": '#007bff' }, 800) });
49
50          $('#table_update_all').click(function(){ //table button to show all NEO data
51                  document.getElementById("table_title").innerHTML = "All Approaches";
52                  var n = 200;
53                  plot_create(n);
54          });
55
56          $('#table_update_top').click(function(){ //table button to show top 10 closest
57                  document.getElementById("table_title").innerHTML = "Top 10 Closest App
58                  var n = 10;
59                  plot_create(n);
60          });
61
62          $("#return").click(topFunction()); //Return to top
63      });
64
65      // Change data content area to visible
66      function display_data(){
67                  $('#data-output').removeClass('data-hidden').addClass('data-display');
68                  $('#return').removeClass('data-hidden'); //reveal back to top link//
69          }
70
71      //Definition of search period function with id as input argument
72      function neo_search_period(id, data_output_state) {
73
74          var state = data_output_state; // define variable based on data output state
75
76          var start_search_date = document.getElementById('initial-search-date').value,
77              start_date = new Date(start_search_date), //create start date variable
78              new_date = new Date(start_date); //create new search date variable
79
80          if (id == 'search') { //if statement based on id returned for search button cl
81
82             search_start(start_search_date, new_date);
83          }
84          else if (id == 'prev') {
85              new_date.setDate(new_date.getDate() - 8); //set new date of search period
86              date_format(new_date); //call function to format date correctly
87              document.getElementById('initial-search-date').value = window.search_date;
88
89              start_search_date = window.search_date;
90              start_date = new Date(start_search_date), //create start date variable
91              new_date = new Date(start_date); //create new search date variable
92
93              if (state == 'true') {
94                  search_start(start_search_date, new_date);
```

CONFIGURE

## Metrics

There are 61 functions in this file.

Function with the largest signature take 3 arguments, while the median is 1.

Largest function has 19 statements in it, while the median is 1.

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

## 29 warnings

- 37 'data_output_state' used out of scope.
- 39 'data_output_state' used out of scope.
- 47 Missing semicolon.
- 48 Missing semicolon.
- 91 Expected an assignment or function call and instead saw an expression.
- 105 Expected an assignment or function call and instead saw an expression.
- 167 ['name'] is better written in dot notation.
- 168 ['nasa_jpl_url'] is better written in dot notation.
- 169 ['close_approach_data'] is better written in dot notation.
- 169 ['close_approach_date'] is better written in dot notation.
- 170 ['absolute_magnitude_h'] is better written in dot notation.
- 171 ['estimated_diameter'] is better written in dot notation.
- 171 ['kilometers'] is better written in dot notation.
- 171 ['estimated_diameter_max'] is better written in dot notation.

```
95              }
96          }
97          else if (id == 'next') {
98
99              new_date.setDate(new_date.getDate() + 8); //set new date of search period
100             date_format(new_date); //call function to format date correctly
101             document.getElementById('initial-search-date').value = window.search_date;
102
103             start_search_date = window.search_date;
104             start_date = new Date(start_search_date), //create start date variable
105             new_date = new Date(start_date); //create new search date variable
106
107             if (state == 'true') {
108                 search_start(start_search_date, new_date);
109             }
110         }
111         else { alert('An error has occured'); } //alert user if error occurs
112     }
113
114     // search function which calls main data generation function
115     function search_start(start, end) {
116             end.setDate(end.getDate() + 7); //set new date of search period
117             date_format(end); //call function to format date correctly
118
119             var search_period = "https://api.nasa.gov/neo/rest/v1/feed?start_date=" +
120                 window.search_date + "&api_key=snyH1wsmtSD133oCQy2spPHmK4PICRb2Y6PdAt4
121
122             retrieve_asteroid_data(search_period, data_extraction, plot_create); //cal
123     }
124
125     // Function to output date in the required format
126     function date_format(date) {
127         window.search_date = date.getFullYear() + "-" + ("0" + (date.getMonth() + 1)).
128     }
129
130
131     //Retrieve data from server and make data available via callback function
132     function retrieve_asteroid_data(search_url, data_create, print) {
133
134         var xhr = new XMLHttpRequest(); //create new XMLHttp request
135
136         xhr.open("GET", search_url); //request data from search URL
137         xhr.send();
138
139         xhr.onreadystatechange = function() {
140             if (this.readyState == 4 && this.status == 200) {
141
142                 data_create(JSON.parse(this.responseText)); //callback function with s
```

CONFIGURE

## Metrics

There are 61 functions in this file.

Function with the largest signature take 3 arguments, while the median is 1.

Largest function has 19 statements in it, while the median is 1.

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

## 29 warnings

- 37 'data_output_state' used out of scope.
- 39 'data_output_state' used out of scope.
- 47 Missing semicolon.
- 48 Missing semicolon.
- 91 Expected an assignment or function call and instead saw an expression.
- 105 Expected an assignment or function call and instead saw an expression.
- 167 ['name'] is better written in dot notation.
- 168 ['nasa_jpl_url'] is better written in dot notation.
- 169 ['close_approach_data'] is better written in dot notation.
- 169 ['close_approach_date'] is better written in dot notation.
- 170 ['absolute_magnitude_h'] is better written in dot notation.
- 171 ['estimated_diameter'] is better written in dot notation.
- 171 ['kilometers'] is better written in dot notation.
- 171 ['estimated_diameter_max'] is better written in dot notation.

JSHint version 2.9.6 (https://github.com/jshint/jshint/releases/tag/2.9.6)

About (/about)
Documentation (/docs)
Install (/install)
Contribute (/contribute)
Blog (/blog)

```
143
144             var n= 10; // define initial table size
145             print(n); //callback function to plot data visualisations
146             display_data(); //display data content area
147         }
148     };
149 }
150
151
152 //function to obtain NEO object data and create new object with required informati
153 function data_extraction(data) {
154
155     window.neo_array = []; //clear previous data from array
156
157     //extract keys from data objects
158     for (var key in data.near_earth_objects) {
159
160         var date = data.near_earth_objects[key]; //obtain keys in 'near_earth_obje
161
162         for (key in date) {
163
164             var neo = date[key]; //obtain keys in 'date' objects
165             var neo_object = {}; //create empty object to store information
166
167             neo_object.name = neo["name"];
168             neo_object.nasa_jpl_url = neo["nasa_jpl_url"];
169             neo_object.close_approach_date = neo["close_approach_data"][0]["close_
170             neo_object.absolute_magnitude_h = neo["absolute_magnitude_h"];
171             neo_object.estimated_diameter_max = neo["estimated_diameter"]["kilomet
172             neo_object.miss_distance_km = neo["close_approach_data"][0]["miss_dist
173             neo_object.relative_velocity_kmps = neo["close_approach_data"][0]["rel
174             neo_object.potential_hazard = neo["is_potentially_hazardous_asteroid"]
175             neo_object.links = neo["links"]["self"];
176
177             window.neo_array.push(neo_object); //push created data objects to arra
178         }
179     }
180 }
181
182 //function to create data visualisation
183 function plot_create(n, display_data) {
184
185     var ndx = crossfilter(window.neo_array); //pass data to crossfilter from NEO ob
186
187     //define chart variables names
188     var neo_object_count_chart, close_approach_stacked, estimated_diameter_stacked
189
190     number_hazardous_objects(ndx, neo_object_count_chart); //call composite chart
```

CONFIGURE

## Metrics

There are 61 functions in this file.

Function with the largest signature take 3 arguments, while the median is 1.

Largest function has 19 statements in it, while the median is 1.

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

## 29 warnings

- 37 'data_output_state' used out of scope.
- 39 'data_output_state' used out of scope.
- 47 Missing semicolon.
- 48 Missing semicolon.
- 91 Expected an assignment or function call and instead saw an expression.
- 105 Expected an assignment or function call and instead saw an expression.
- 167 ['name'] is better written in dot notation.
- 168 ['nasa_jpl_url'] is better written in dot notation.
- 169 ['close_approach_data'] is better written in dot notation.
- 169 ['close_approach_date'] is better written in dot notation.
- 170 ['absolute_magnitude_h'] is better written in dot notation.
- 171 ['estimated_diameter'] is better written in dot notation.
- 171 ['kilometers'] is better written in dot notation.
- 171 ['estimated_diameter_max'] is better written in dot notation.

JSHint version 2.9.6 (https://github.com/jshint/jshint/releases/tag/2.9.6)

About (/about)
Documentation (/docs)
Install (/install)
Contribute (/contribute)
Blog (/blog)

```
191       close_approach_stack(ndx, close_approach_stacked); //call close approach stack
192       estimated_diameter_stack(ndx, estimated_diameter_stacked); //call estimated di
193       potential_hazard(ndx, hazardous_neo); //call potential hazard pie chart
194       neo_count(ndx, total_neo_count); //call total count function
195       neo_data_table(ndx, neo_object_table, n); //call create NEO data table functio
196
197       dc.renderAll(); //render all plots
198   }
199
200   //miss distance function
201   function miss_distance(dimension, min_distance, max_distance) {
202       return dimension.group().reduce(
203           function(p, v) {
204               if (min_distance <= v.miss_distance_km) {
205                   if (v.miss_distance_km < max_distance) {
206                       p.total++;
207                   }
208               }
209               return p;
210           },
211           function(p, v) {
212               if (min_distance <= v.miss_distance_km) {
213                   if (v.miss_distance_km < max_distance) {
214                       p.total--;
215                   }
216               }
217               return p;
218           },
219           function() {
220               return { total: 0 };
221           }
222       );
223   }
224
225   //estimated diameter function
226   function estimated_diameter(dimension, min_size, max_size) {
227       return dimension.group().reduce(
228           function(p, v) {
229               if (min_size <= v.estimated_diameter_max) {
230                   if (v.estimated_diameter_max < max_size) {
231                       p.total++;
232                   }
233               }
234               return p;
235           },
236           function(p, v) {
237               if (min_size <= v.estimated_diameter_max) {
238                   if (v.estimated_diameter_max < max_size) {
```

CONFIGURE

## Metrics

There are 61 functions in this file.

Function with the largest signature take 3 arguments, while the median is 1.

Largest function has 19 statements in it, while the median is 1.

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

## 29 warnings

- 37 'data_output_state' used out of scope.
- 39 'data_output_state' used out of scope.
- 47 Missing semicolon.
- 48 Missing semicolon.
- 91 Expected an assignment or function call and instead saw an expression.
- 105 Expected an assignment or function call and instead saw an expression.
- 167 ['name'] is better written in dot notation.
- 168 ['nasa_jpl_url'] is better written in dot notation.
- 169 ['close_approach_data'] is better written in dot notation.
- 169 ['close_approach_date'] is better written in dot notation.
- 170 ['absolute_magnitude_h'] is better written in dot notation.
- 171 ['estimated_diameter'] is better written in dot notation.
- 171 ['kilometers'] is better written in dot notation.
- 171 ['estimated_diameter_max'] is better written in dot notation.

JSHint version 2.9.6 (https://github.com/jshint/jshint/releases/tag/2.9.6)

About (/about)
Documentation (/docs)
Install (/install)
Contribute (/contribute)
Blog (/blog)

```
239                     p.total--;
240                 }
241             }
242             return p;
243         },
244         function() {
245             return { total: 0 };
246         }
247     );
248 }
249
250 //NEO objects line chart
251 function number_hazardous_objects(data, chart) {
252
253     var parse_date = d3.time.format("%Y-%m-%d").parse; //set format for date strin
254
255     window.neo_array.forEach(function(d) {
256         d.date = parse_date(d.close_approach_date); //parse date string
257     });
258
259     var close_approach_date_dim = data.dimension(function(d) { //create dimension
260         return d.date;
261     });
262
263     var min_date = close_approach_date_dim.bottom(1)[0].date; //create start date
264     var max_date = close_approach_date_dim.top(1)[0].date; //create end date
265
266     var close_approaches = close_approach_date_dim.group().reduce( //create functi
267         function reduceAdd(p, v) { return p + 1; },
268         function reduceRemove(p, v) { return p - 1; },
269         function reduceInitialise() { return 0; }
270     );
271
272     var hazards = close_approach_date_dim.group().reduceSum(function(d) { //create
273         if (d.potential_hazard === true) { return +d.potential_hazard; }
274         else { return 0; }
275     });
276
277     var lessThan10millionkm = miss_distance(close_approach_date_dim, 0, 10); //cal
278     var greaterThan2km = estimated_diameter(close_approach_date_dim, 2, 10); //cal
279
280     chart = dc.compositeChart("#neo-count"); //bind data to html element
281     chart //create chart
282         .margins({ top: 60, right: 30, bottom: 80, left: 40 })
283         .dimension(close_approach_date_dim)
284         .width(550)
285         .height(350)
286         .x(d3.time.scale().domain([min_date, max_date]))
```

CONFIGURE

## Metrics

There are 61 functions in this file.

Function with the largest signature take 3 arguments, while the median is 1.

Largest function has 19 statements in it, while the median is 1.

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

## 29 warnings

- 37 'data_output_state' used out of scope.
- 39 'data_output_state' used out of scope.
- 47 Missing semicolon.
- 48 Missing semicolon.
- 91 Expected an assignment or function call and instead saw an expression.
- 105 Expected an assignment or function call and instead saw an expression.
- 167 ['name'] is better written in dot notation.
- 168 ['nasa_jpl_url'] is better written in dot notation.
- 169 ['close_approach_data'] is better written in dot notation.
- 169 ['close_approach_date'] is better written in dot notation.
- 170 ['absolute_magnitude_h'] is better written in dot notation.
- 171 ['estimated_diameter'] is better written in dot notation.
- 171 ['kilometers'] is better written in dot notation.
- 171 ['estimated_diameter_max'] is better written in dot notation.

JSHint version 2.9.6 (https://github.com/jshint/jshint/releases/tag/2.9.6)

About (/about)
Documentation (/docs)
Install (/install)
Contribute (/contribute)
Blog (/blog)

```
287           .shareTitle(false)
288           .compose([
289               dc.lineChart(chart)
290               .colors('#96bae2')
291               .group(close_approaches, "Total number of NEO's per day").title(functi
292               dc.lineChart(chart)
293               .colors('#eb5d5d')
294               .group(hazards, "Neo's potentially hazardous to Earth").title(function
295               dc.lineChart(chart)
296               .colors('#ade49b')
297               .group(lessThan10millionkm, "Miss distance less than 10 million km ")
298               .valueAccessor(function(d) {
299                   if (d.value.total > 0) { return d.value.total; }
300                   else { return 0; }
301               }).title(function(d) { return "There are a total of " + d.value.total
302               dc.lineChart(chart)
303               .colors('#e378e4')
304               .group(greaterThan2km, "Estimated diameter greater than 2km")
305               .valueAccessor(function(d) {
306                   if (d.value.total > 0) { return d.value.total; }
307                   else { return 0; }
308               }).title(function(d) { return "There are a total of " + d.value.total
309           ])
310
311           .brushOn(false)
312           .elasticX(true)
313           .elasticY(true)
314           .xAxisLabel("Close approach date")
315           .yAxisLabel("Number of NEO objects")
316           .legend(dc.legend().x(60).y(-2).itemWidth(250).gap(20).horizontal(true))
317           .render()
318           .xAxis().tickFormat(d3.time.format("%Y-%m-%d")).ticks(8);
319
320   }
321
322   //Close approach distance stacked chart
323   function close_approach_stack(data, chart) {
324
325       var close_approach_date_dim = data.dimension(dc.pluck('close_approach_date'));
326
327       //call miss distance function with required arguments
328       var lessThan10millionkm = miss_distance(close_approach_date_dim, 0, 10);
329       var lessThan50millionkm = miss_distance(close_approach_date_dim, 10, 50);
330       var moreThan50millionkm = miss_distance(close_approach_date_dim, 50, 100);
331
332       chart = dc.barChart("#close-approach-plot"); //bind data to stacked chart
333       chart
334           .margins({ top: 50, right: 30, bottom: 80, left: 40 })
```

CONFIGURE

## Metrics

There are 61 functions in this file.

Function with the largest signature take 3 arguments, while the median is 1.

Largest function has 19 statements in it, while the median is 1.

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

## 29 warnings

- 37 'data_output_state' used out of scope.
- 39 'data_output_state' used out of scope.
- 47 Missing semicolon.
- 48 Missing semicolon.
- 91 Expected an assignment or function call and instead saw an expression.
- 105 Expected an assignment or function call and instead saw an expression.
- 167 ['name'] is better written in dot notation.
- 168 ['nasa_jpl_url'] is better written in dot notation.
- 169 ['close_approach_data'] is better written in dot notation.
- 169 ['close_approach_date'] is better written in dot notation.
- 170 ['absolute_magnitude_h'] is better written in dot notation.
- 171 ['estimated_diameter'] is better written in dot notation.
- 171 ['kilometers'] is better written in dot notation.
- 171 ['estimated_diameter_max'] is better written in dot notation.

```
335         .width(500)
336         .height(350)
337         .dimension(close_approach_date_dim)
338         .group(lessThan10millionkm, "less than 10*10^6 km")
339         .stack(lessThan50millionkm, "less than 50*10^6 km")
340         .stack(moreThan50millionkm, "greater than 50*10^6 km")
341         .valueAccessor(function(d) {
342             if (d.value.total > 0) { return d.value.total; }
343             else { return 0; }
344         })
345         .x(d3.scale.ordinal())
346         .xUnits(dc.units.ordinal)
347         .ordinalColors(['#eb5d5d', '#b7cee8', '#ade49b'])
348         .title('less than 10*10^6 km', function(d) { return 'There are ' + d.value
349         .title('less than 50*10^6 km', function(d) { return 'There are ' + d.value
350         .title('greater than 50*10^6 km', function(d) { return 'There are ' + d.va
351         .legend(dc.legend().x(50).y(-2).itemWidth(140).gap(5).horizontal(true))
352         .yAxisLabel("Number of NEO objects")
353         .xAxisLabel("Close approach date");
354 }
355
356 //Estimated maximum diameter stacked chart function
357 function estimated_diameter_stack(data, chart) {
358
359     var close_approach_date_dim = data.dimension(dc.pluck('close_approach_date'));
360
361     //call estimated diameter function with arguments
362     var lessThan1km = estimated_diameter(close_approach_date_dim, 0, 1);
363     var lessThan2km = estimated_diameter(close_approach_date_dim, 1, 2);
364     var moreThan2km = estimated_diameter(close_approach_date_dim, 2, 100);
365
366     chart = dc.barChart("#estimated-diameter-stack"); //bind data to stacked chart
367     chart
368         .margins({ top: 50, right: 30, bottom: 80, left: 40 })
369         .width(500)
370         .height(350)
371         .dimension(close_approach_date_dim)
372         .group(lessThan1km, "less than 1km")
373         .stack(lessThan2km, "less than 2km")
374         .stack(moreThan2km, "greater than 2km")
375         .valueAccessor(function(d) {
376             if (d.value.total > 0) { return d.value.total; }
377             else { return 0; }
378         })
379         .x(d3.scale.ordinal())
380         .xUnits(dc.units.ordinal)
381         .ordinalColors(['#ade49b', '#b7cee8', '#eb5d5d'])
382         .title('less than 1km', function(d) { return 'There are ' + d.value.total
```

CONFIGURE

## Metrics

There are 61 functions in this file.

Function with the largest signature take 3 arguments, while the median is 1.

Largest function has 19 statements in it, while the median is 1.

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

## 29 warnings

- 37 'data_output_state' used out of scope.
- 39 'data_output_state' used out of scope.
- 47 Missing semicolon.
- 48 Missing semicolon.
- 91 Expected an assignment or function call and instead saw an expression.
- 105 Expected an assignment or function call and instead saw an expression.
- 167 ['name'] is better written in dot notation.
- 168 ['nasa_jpl_url'] is better written in dot notation.
- 169 ['close_approach_data'] is better written in dot notation.
- 169 ['close_approach_date'] is better written in dot notation.
- 170 ['absolute_magnitude_h'] is better written in dot notation.
- 171 ['estimated_diameter'] is better written in dot notation.
- 171 ['kilometers'] is better written in dot notation.
- 171 ['estimated_diameter_max'] is better written in dot notation.

JS Hint

version 2.9.6 (https://github.com/jshint/jshint/releases/tag/2.9.6)

```
383           .title('less than 2km', function(d) { return 'There are ' + d.value.total
384           .title('greater than 2km', function(d) { return 'There are ' + d.value.tot
385           .legend(dc.legend().x(100).y(-2).itemWidth(100).gap(5).horizontal(true))
386           .yAxisLabel("Number of NEO objects")
387           .xAxisLabel("Close approach date");
388   }
389
390   //Potential hazard pie chart
391   function potential_hazard(data, chart) {
392
393       var hazard_size = data.dimension(function(d) { //create dimension based on pot
394           if (d.potential_hazard == true) { return 'YES'; }
395           else { return 'NO'; }
396       });
397
398       var hazard_size_group = hazard_size.group(); //create data group
399
400       chart = dc.pieChart("#potential-hazard"); //bind data to chart
401       chart
402           .radius(130)
403           .innerRadius(50)
404           .height(400)
405           .width(300)
406           .dimension(hazard_size)
407           .group(hazard_size_group)
408           .ordinalColors(['#ade49b', '#eb5d5d'])
409           .title(function(d) {
410               if (d.key === 'YES') {
411                   return "A potential risk to Earth is posed by these " + d.value +
412               }
413               else { return "No risk to Earth is posed by these " + d.value + " NEO'
414           })
415           .renderLabel(true)
416           .legend(dc.legend().x(110).y(350).itemWidth(60).gap(5).horizontal(true))
417           .transitionDuration(500);
418   }
419
420   //Total object count
421   function neo_count(data, count) {
422
423       var all = data.groupAll();
424
425       count = dc.dataCount("#data-count")
426           .dimension(data)
427           .group(all);
428   }
429
430   //NEO data table
```

CONFIGURE

## Metrics

There are 61 functions in this file.

Function with the largest signature take 3 arguments, while the median is 1.

Largest function has 19 statements in it, while the median is 1.

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

## 29 warnings

- 37 'data_output_state' used out of scope.
- 39 'data_output_state' used out of scope.
- 47 Missing semicolon.
- 48 Missing semicolon.
- 91 Expected an assignment or function call and instead saw an expression.
- 105 Expected an assignment or function call and instead saw an expression.
- 167 ['name'] is better written in dot notation.
- 168 ['nasa_jpl_url'] is better written in dot notation.
- 169 ['close_approach_data'] is better written in dot notation.
- 169 ['close_approach_date'] is better written in dot notation.
- 170 ['absolute_magnitude_h'] is better written in dot notation.
- 171 ['estimated_diameter'] is better written in dot notation.
- 171 ['kilometers'] is better written in dot notation.
- 171 ['estimated_diameter_max'] is better written in dot notation.

JSHint

version 2.9.6 (https://github.com/jshint/jshint/releases/tag/2.9.6)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

```
431  function neo_data_table(data, table, n) {
432
433      var neo_data_dim = data.dimension(function(d) { return d; }); //create dimensi
434
435      table = dc.dataTable('#neo_data_table'); //bind data to table
436      table
437          .dimension(neo_data_dim)
438          .group(function(d) { return d; })
439          .size(n)
440          .columns([
441              function(d) { return d.close_approach_date; },
442              function(d) { return d.name; },
443              function(d) { return d.miss_distance_km.toPrecision(4); },
444              function(d) { return d.estimated_diameter_max.toPrecision(4); },
445              function(d) { return d.potential_hazard; },
446              function(d) { return '<a id="neo-link" href=\"' + d.nasa_jpl_url + '\"
447          ])
448          .sortBy(function(d) { return d.miss_distance_km; })
449          .order(d3.ascending)
450          .on('renderlet', function(table) {
451              table.select('tr.dc-table-group').remove();
452          });
453  }
```

CONFIGURE

## Metrics

There are 61 functions in this file.

Function with the largest signature take 3 arguments, while the median is 1.

Largest function has 19 statements in it, while the median is 1.

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

## 29 warnings

- 37 'data_output_state' used out of scope.
- 39 'data_output_state' used out of scope.
- 47 Missing semicolon.
- 48 Missing semicolon.
- 91 Expected an assignment or function call and instead saw an expression.
- 105 Expected an assignment or function call and instead saw an expression.
- 167 ['name'] is better written in dot notation.
- 168 ['nasa_jpl_url'] is better written in dot notation.
- 169 ['close_approach_data'] is better written in dot notation.
- 169 ['close_approach_date'] is better written in dot notation.
- 170 ['absolute_magnitude_h'] is better written in dot notation.
- 171 ['estimated_diameter'] is better written in dot notation.
- 171 ['kilometers'] is better written in dot notation.
- 171 ['estimated_diameter_max'] is better written in dot notation.

CONFIGURE


version 2.9.6
(https://github.com
/jshint/jshint/releases
/tag/2.9.6)

About (/about)
Documentation (/docs)
Install (/install)
Contribute (/contribute)
Blog (/blog)

## Metrics

There are 61 functions in this file.

Function with the largest signature take 3 arguments, while the median is 1.

Largest function has 19 statements in it, while the median is 1.

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

## 29 warnings

| | |
|---|---|
| 37 | 'data_output_state' used out of scope. |
| 39 | 'data_output_state' used out of scope. |
| 47 | Missing semicolon. |
| 48 | Missing semicolon. |
| 91 | Expected an assignment or function call and instead saw an expression. |
| 105 | Expected an assignment or function call and instead saw an expression. |
| 167 | ['name'] is better written in dot notation. |
| 168 | ['nasa_jpl_url'] is better written in dot notation. |
| 169 | ['close_approach_data'] is better written in dot notation. |
| 169 | ['close_approach_date'] is better written in dot notation. |
| 170 | ['absolute_magnitude_h'] is better written in dot notation. |
| 171 | ['estimated_diameter'] is better written in dot notation. |
| 171 | ['kilometers'] is better written in dot notation. |
| 171 | ['estimated_diameter_max'] is better written in dot notation. |